



Universidad
Carlos III de Madrid

Departamento de Informática

TRABAJO FIN DE GRADO

BookSelling

Aplicación web de compra-venta de libros de segunda mano

Autor: Jorge Alonso López

Tutor: Jesús Hernando Corrochano

Título: Aplicación web de compra-venta de libros de segunda mano (BookSelling)

Autor: Jorge Alonso López

Tutor: Jesús Hernando Corrochano

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Me gustaría dar las gracias en primer lugar a mi madre, mi padre, a mi hermana y a mis “yayis” ya que todos ellos son los que me han permitido y se han sacrificado para que yo esté aquí. Jamás podré devolver toda la confianza puesta en mí y en especial agradecer a mi padre que siempre donde esté me envía las ganas y las fuerzas de seguir hacia adelante.

También agradecer a toda la gente que he conocido a lo largo de la carrera, que han llenado de grandes momentos mi vida, dentro y fuera de la Universidad.

Mención especial a mi gran amigo Rubén que ha sido mi gran apoyo ante todos los problemas que me han surgido y que gracias a sus consejos esto ha ido siempre hacia delante.

No podía faltar mi agradecimiento a mi pareja Marta, que me ha apoyado siempre y ha confiado en mí para sacar esto adelante. Sin su apoyo no habría conseguido nada de esto.

Por último agradecer a mi tutor Jesús Hernando por darme la oportunidad de desarrollar este proyecto con él, y por sus consejos y ayuda en todo momento.

Gracias a todas las persona que me han ayudado o contribuido a la realización de este proyecto.

Muchas gracias.

Resumen

El acceso a todos los artículos existentes en el mercado no siempre está al alcance de todas las personas, ya que existen diferentes niveles económicos. Éste problema abre la posibilidad de explotar un gran sector económico como es la comercialización de artículos de segunda mano. Esto ha hecho que muchas personas utilicen la web, como escaparate de esos artículos usados permitiéndoles venderlos a precios más asequibles.

El problema que resuelve este proyecto es la creación de una aplicación web que ayude a la compra-venta de artículos de segunda mano de forma exclusiva, dedicando todo su énfasis en la venta de un producto en particular, en este caso libros usados.

Este problema junto con la importancia que está alcanzando el uso de las tecnologías móviles y el aumento del uso de éstas, ha hecho que se piense en global a la hora de desarrollar esta aplicación, intentando que pueda ser utilizada en todo tipo de aparatos móviles (Smartphone, Tablet, etc.).

El objetivo fundamental de este proyecto es diseñar una tienda online de compra-venta de libros usados adaptable a cualquier dispositivo electrónico con acceso a internet.

La metodología de trabajo ha sido el desarrollo en cascada, pero con una característica que los requisitos no están cerrados desde un principio, sino que pueden sufrir modificaciones o simplemente que pueden aparecer nuevos.

Se ha elegido Node.js como plataforma de desarrollo y se han utilizado varias herramientas para su desarrollo, como son el lenguaje de programación JavaScript [16] y el framework Bootstrap.

Como resultado se ha obtenido una aplicación web que permite a sus usuarios concretar pedidos de libros de otros usuarios de la web. Los usuarios podrán navegar por el catalogo, descubrir las diferentes categorías de libros existentes, realizar pedidos de libros y buscar sus títulos preferidos.

Palabras clave: libros usados, aplicaciones e-commerce, Node.

Abstract

Due to the differences in the economic levels of the population not everybody can access to everything is sold in the markets. This issue has facilitated the expansion of a second hand market. The web enables people to find these products more easily and enables the seller to reduce the prices making these products more affordable to the public.

With this project we have tried to create a web application that helps in the process of selling-buying these second hand products, mainly dedicated to second hand books.

This issue together with the importance of the mobile technologies have prompted us to think global while developing this application so it can be used in several devices (smartphones, tablets, etc).

The main aim of this project is the design of an online shop for buying-selling second hand books that can be used in any device with web access.

The methodology used has been the waterfall development but with a difference, the requirements are not closed from the beginning but these can be modified or new ones can appear.

We have chosen Node.js as the platform for the development and we have used several tools like the JavaScript [16] programming language and the Bootstrap framework.

As a result we have obtained a web application that allows its user to arrange orders with other users. The users can view the catalogue, look through different books categories, order books, and find their favourite titles.

Keywords: second hand books, e-commerce applications, Node.

Índice

1. CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS 14

1.1 Motivación del proyecto 14

1.2 Objetivos 15

1.3 Motivación 15

1.4 Que se espera con este proyecto..... 16

1.5 Medios empleados..... 16

1.5.1 Elementos hardware 16

1.5.2 Elementos software 17

1.6 Terminología 17

1.6.1 Acrónimos y abreviaturas 17

1.6.2 Definiciones..... 18

1.7 Estructura de la memoria..... 19

2. CAPÍTULO 2. ESTADO DEL ARTE 21

2.1 Criterios de búsqueda 21

2.2 Análisis de las aplicaciones existentes en el mercado 22

2.2.1 AbeBooks 22

2.2.2 La casa del libro 24

2.2.3 Amazon 25

2.3 Análisis de alternativas tecnológicas 25

2.3.1 Alternativa 1 – Desarrollo aplicación web con JEE 26

2.3.2 Alternativa 2 - Desarrollo aplicación web con Node.js 27

2.4 Análisis de metodologías para realizar el proyecto 29

2.4.1 Metodologías ágiles 29

2.4.2 The Agile Alliance 30

2.4.3 Scrum 30

2.4.4 Metodología tradicional..... 32

2.4.4.1 RUP (Rational Unified Procces) 32

2.4.4.2 Modelo en cascada. 35

2.4.4.3 Comparativa entre metodologías ágiles y tradicionales..... 37

2.5 Conclusiones 37

2.6 Solución propuesta 39

2.6.1 Entorno tecnológico seleccionado 39

3.	CAPÍTULO 3. ANÁLISIS DE REQUISITOS	46
3.1	Descripción general.....	46
3.2	Requisitos de usuario.....	47
3.2.1	Requisitos de capacidad.....	48
3.2.2	Requisitos de restricción	52
3.3	Casos de uso.....	54
3.3.1	Diagramas UML	54
3.3.2	Descripción contextual.....	56
3.4	Requisitos de software.....	66
3.4.1	Requisitos funcionales	67
3.4.2.	Requisitos no funcionales	74
4.	CAPÍTULO 4 DISEÑO	76
4.1.	Introducción	76
4.2.	Diseño: Arquitectura	76
4.2.1.	View.....	78
4.2.2.	Controller	78
4.2.3.	Model	81
4.3.	Implementación: Interfaz web final.....	84
4.3.1.	Interfaz usuario	84
4.3.1.1.	Inicio	84
4.3.1.2.	Menú personal	85
4.3.1.3.	Buscador.....	86
4.3.1.4.	Acceso a la web.....	87
4.3.1.5.	Registro en la web.....	88
4.3.1.6.	Libros.....	89
4.3.1.7.	Contacto	90
4.3.2.	Interfaz usuario administrador	91
4.3.2.1.	Interfaz Usuario administrador.....	91
4.3.3.	Interfaz: Multi-dispositivo	92
4.3.3.1.	Interfaz Smartphone	92
4.3.3.2.	Interfaz Tablet	92
5.	Capítulo 5. Plan de pruebas	93
5.1	Pruebas de aceptación	93
5.1.1	Resultado de la pruebas.....	102
6.	CAPÍTULO 6 GESTIÓN DEL PROYECTO.....	103

- 6.1 Planificación del proyecto103
 - 6.1.1 Planificación inicial103
 - 6.1.2 Planificación final105
- 6.2 Análisis económico.....107
 - 6.2.1 Coste personal.....107
 - 6.2.2 Coste del Hardware.....108
 - 6.2.3 Coste del software108
 - 6.2.4 Costes totales.....109
- 7. CAPÍTULO 7: CONCLUSIONES.....110
 - 7.1. Conclusiones del proyecto110
 - 7.2. Líneas futuras111
- 8. BIBLIOGRAFÍA.....113

Índice de tablas.

Tabla 1. Diferencias entre Metodología ágil y metodología tradicional.....	37
Tabla 2: Comparativa características web.	38
Tabla 3. Plantilla requisitos de usuario	47
Tabla 4. RUC-01. Registro de usuario.....	48
Tabla 5. RUC-02: Acceso a la web.	48
Tabla 6. RUC-03. Acceso menú personal.	49
Tabla 7. RUC-04. Buscar libro.....	49
Tabla 8. RUC-05. Visualizar catálogo.....	50
Tabla 9. RUC-06. Visualizar libro disponible	50
Tabla 10. RUC-07. Ordenar libros.	51
Tabla 11. RUC-08. Pre-visualizar carrito.....	51
Tabla 12. RUC-09. Mensaje de contacto.....	52
Tabla 13. RUC-10. Salir de la web.	52
Tabla 14. RUR-01. Idioma.....	52
Tabla 15. RUR-02. Resolución interfaz web.	53
Tabla 16. RUR-03. Navegador	53
Tabla 17. RUR-04. Interfaz.	53
Tabla 18. Plantilla casos de uso genérico.....	56
Tabla 19. CUA-01. Gestionar usuarios.	57
Tabla 20. CUA-02. Gestionar libros.	57
Tabla 21. CUU-01. Registrarse.	58
Tabla 22. CUU-02. Usuario conectado.	58
Tabla 23. CUU-03. Usuario no conectado.	59
Tabla 24. CUU-04. Buscar libros.....	59
Tabla 25. CUU-05. Ver catálogo libros disponibles.	59
Tabla 26. CUU-06. Ver detalles libro.	60
Tabla 27. CUU-07. Ordenar libros.	60
Tabla 28. CUU-08. Acceso menú personal.	61
Tabla 29. CUU-09. Ver datos personales.	61
Tabla 30. CUU-10. Modificar datos personales.	62
Tabla 31. CUU-11. Ver listado pedidos.	62
Tabla 32. CUU-12. Ver carro pedido actual.	63
Tabla 33. CUU-13. Subir libro.....	63
Tabla 34. CUU-14. Mis libros.....	64
Tabla 35. CUU-15. Añadir libro al carro.	64
Tabla 36. CUU-16. Eliminar libro del carro.....	65
Tabla 37. CUU-17. Enviar mensaje de consulta.	65
Tabla 38. CUU-18. Cerrar sesión.	66
Tabla 39. Plantilla RSF y RSNF.	66
Tabla 40. RSF-01. Registro de usuario.....	67
Tabla 41. RSF-02. Acceso a la web.	67
Tabla 42. RSF-03. Accesos a los datos personales.	68

Tabla 43. RSSF-04. Modificar datos personales.	68
Tabla 44. RSF-05. Subir libros.	68
Tabla 45. RSF-06. Historial de libros subidos.	69
Tabla 46. RSF-07. Historial de pedidos.	69
Tabla 47. RSF-08. Carrito de pedido actual.	69
Tabla 48. RSF-09. Buscar libro.	70
Tabla 49. RSF-10. Libros por categorías.	70
Tabla 50. RSF-11. Top ventas.	70
Tabla 51. RSF-12. Novedades.	71
Tabla 52. RSF-13. Ver detalles del libro.	71
Tabla 53. RSF-14. Añadir libro al carro del pedido.	71
Tabla 54. RSF-15. Ordenar libros alfabéticamente I.	72
Tabla 55. RSF-16. Ordenar libros alfabéticamente II.	72
Tabla 56. RSF-17. Ordenar libros por precio I.	72
Tabla 57. RSF-18. Ordenar libros por precio II.	73
Tabla 58. RSF-19. Eliminar libro del carrito.	73
Tabla 59. RSF-20. Envío de mensaje de contacto.	73
Tabla 60. RSF-21. Salir de la web.	74
Tabla 61. Acrónimos Requisitos No funcionales.	74
Tabla 62. RSNFI-01. Idioma.	74
Tabla 63. RSNFI-02. Resolución interfaz de la web.	75
Tabla 64. RSNFI-03. Interfaz.	75
Tabla 65. RSNFO-04. Navegadores.	75
Tabla 66. Plantilla prueba aceptación.	93
Tabla 67. PA-01. Registro de usuario.	93
Tabla 68. PA-02. Acceso a la web.	94
Tabla 69. PA-03. Acceso a los datos personales.	94
Tabla 70. PA-04. Modificar datos personales.	94
Tabla 71. PA-05. Subir libros.	95
Tabla 72. PA-06. Historial de libros subidos.	95
Tabla 73. PA-07. Historial de pedidos.	95
Tabla 74. PA-08. Carrito de pedido actual.	96
Tabla 75. PA-09. Buscar libro.	96
Tabla 76. PA-10. Libros por categorías.	96
Tabla 77. PA-11. Top ventas.	97
Tabla 78. PA-12. Novedades.	97
Tabla 79. PA-13. Ver detalles del libro.	97
Tabla 80. PA-14. Añadir libro al carro del pedido.	98
Tabla 81. PA-15. Ordenar libros alfabéticamente (a-z).	98
Tabla 82. PA-16. Ordenar alfabéticamente (z-a).	98
Tabla 83. PA-17. Ordenar libros por precio ascendente.	99
Tabla 84. PA-18. Ordenar libros por precio descendente.	99
Tabla 85. PA-19. Eliminar libro del carrito.	99
Tabla 86. PA-20. Envío mensaje de contacto.	100
Tabla 87. PA-21. Salir de la web.	100
Tabla 88. Matriz de trazabilidad. Prueba aceptación – Requisito funcional.	101

Tabla 89. Resultado de las pruebas.102

Tabla 90. Detalles coste personal.107

Tabla 91. Coste de hardware.108

Tabla 92. Coste del software.....108

Tabla 93. Costes totales.109

Índice de ilustraciones.

Ilustración 1: Logo AbeBooks	22
Ilustración 2: Funcionamiento AbeBooks	22
Ilustración 3: Ejemplo web AbeBooks.....	23
Ilustración 4: Logo La casa del libro	24
Ilustración 5: Ejemplo web Casa del libro	24
Ilustración 6: Logo de Amazon	25
Ilustración 7: Ejemplo web Amazon.....	25
Ilustración 8: Logo Node.js.....	27
Ilustración 9: Comparativa Java vs Node.js.....	28
Ilustración 10. Metodología SCRUM.	32
Ilustración 11. Resumen estructura dinámica RUP.....	35
Ilustración 12. Fases del método en cascada.....	35
Ilustración 13: Logo de Bootstrap.	40
Ilustración 14: Lenguajes programación Bootstrap.	41
Ilustración 15: Logo MySQL.....	41
Ilustración 16: Logo MongoDB.....	42
Ilustración 17: Logo de Sublime Text 3	43
Ilustración 18: Logo de GitHub.	44
Ilustración 19. Casos de uso, usuario Administrador.....	54
Ilustración 20. Casos de uso. Usuario	55
Ilustración 21. Arquitectura Model-View-Controller	76
Ilustración 22. Esquema de la arquitectura diseñada.....	77
Ilustración 23. Controlador de BookSelling.....	78
Ilustración 24. Servicios, vistas y módulos.....	81
Ilustración 25: Modelo entidad-relacion. BookSelling.....	82
Ilustración 26: Sesión y carrito en MongoDB.....	83
Ilustración 27: Página de inicio de la aplicación.	84
Ilustración 28. Menú personal del usuario.	85
Ilustración 29. Buscador de libros de aplicación.....	86
Ilustración 30. Página de acceso a la web.....	87
Ilustración 31. Página de registro en la web.....	88
Ilustración 32. Libros mostrados por la aplicación.	89
Ilustración 33: Página de contacto de la aplicación	90
Ilustración 34. Administración. Gestión de usuarios.	91
Ilustración 35. Administración. Gestión de los libros.	91
Ilustración 36. Interfaz ejemplo en Smartphone	92
Ilustración 37. Interfaz ejemplo en una tablet.....	92
Ilustración 38. Planificación inicial del proyecto.....	104
Ilustración 39. Planificación final del proyecto.....	106

1. CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS

En este capítulo se presenta el contexto del proyecto, la motivación que ha llevado a su desarrollo, sus objetivos, los medios empleados y la estructura del resto de la memoria.

1.1 Motivación del proyecto

Las últimas décadas constituyen una época en la que se han sucedido una cantidad tal de innovaciones tecnológicas y científicas que algunos autores han concluido en denominarla la “tercera revolución industrial” [35]. Entre los sectores más significativos de este avance se encuentran la energía nuclear, la robótica, la informática o las telecomunicaciones. Todos ellos han impulsado a la sociedad hacia un profundo cambio económico y social.

Especial relevancia en cuanto al impacto social adquiere la evolución en el ámbito de las tecnologías de la información y la comunicación, con repercusiones tanto a nivel personal, como empresarial o político. El desarrollo de satélites artificiales de comunicaciones, microprocesadores, ordenadores, teléfonos móviles o Internet, entre otras tecnologías, ha cambiado radicalmente la forma en que la se obtiene y emite información, permitiendo una comunicación instantánea y eficaz desde cualquier lugar y en cualquier momento. También, han abierto la puerta a un acceso a la información mucho más amplio y rico, retroalimentando aún más si cabe la evolución tecnológica.

En este contexto en el que la forma de vida está cada vez más integrada con la tecnología y las nuevas generaciones han crecido con ella, parece bastante lógico pensar que un ámbito tan importante en la sociedad, como es el comercio, evolucione de la mano. Surge así el comercio electrónico, basado en el uso de las nuevas tecnologías para realizar transacciones, en definitiva, el pago por algo tangible a través de medios electrónicos.

Dentro del e-commerce, una de las ramas con más futuro y que más ha crecido en las últimas décadas, es la orientada a la compra-venta de bienes de segunda mano o seminuevos [33]. En la actualidad muchas personas utilizan el duro momento económico que atraviesa nuestro país, para iniciar su negocio a través de internet. Muchos de ellos continuando con lo que ya realizaban de forma física, pero el avance de las tecnologías les ha hecho dar el salto de manera casi obligada.

Gracias al gran avance existente en el mundo de las nuevas tecnologías, cada vez más gente utiliza otros dispositivos para navegar por Internet, ya sean Smartphone o Tablet, cada uno de ellos con resolución diferente debido a su tamaño. Por tanto toda aplicación web creada en la actualidad de e-commerce deberá poder adaptarse a estos nuevos dispositivos. Este hecho no implica una pérdida de funcionalidad o de navegabilidad. Según estudios toda aplicación e-commerce existente en la actualidad deberá ser portable a cada uno de estos aparatos existentes en el mundo digital para poder satisfacer las tendencias de los usuarios sobre estos aparatos.

En este contexto el presente proyecto se enmarca dentro del desarrollo de una aplicación web que trata de aliviar esta situación permitiendo a los consumidores tener a su alcance utilizando el

dispositivo que deseen, una serie de productos, en este caso libros, y a los vendedores ofrecerles un lugar donde poder ofrecerlos, facilitando el intercambios entre ambas partes.

1.2 Objetivos

El objetivo fundamental de este proyecto es el de diseñar e implementar una aplicación web de compra-venta de libros usados, adaptable a cualquier dispositivo electrónico con acceso a internet, utilizando la tecnología Node.js. En base a este objetivo principal, se proponen como objetivos secundarios:

- La aplicación debe atraer la atención de los usuarios

Este objetivo es esencial, porque existen grandes potencias en el mundo de las tecnologías que ofrecen este tipo de servicios, además de la compra de artículos nuevos, por tanto debe captar la atención del usuario que acceda a ella.

- La aplicación debe ser sencilla y de fácil comprensión

De nada sirve captar la atención del usuario, si después el usuario no sabe utilizar ni realizar nada en la web.

- La aplicación a ser posible que sea original

Este objetivo es menos importante que los anteriores, pero no se debe caer en la rutina o en lo normal, sino sorprender desde el primer momento al usuario.

1.3 Motivación

La motivación personal a la hora de decidir abordar la realización de este proyecto ha venido especialmente fundamentada en las siguientes razones:

- **Investigación y aplicación de nuevas tecnologías emergentes en el e-commerce.**

Se puede considerar el motivo principal. La industria del comercio online vive en cambio constante debido a la aparición de nuevos dispositivos, nuevas tecnologías de desarrollo de aplicación, y tienen que estar preparados para su adaptación. Esto hace que exista una gran variedad de posibilidades a la hora de desarrollar una aplicación web. Es por ello que el hecho de realizar un estudio sobre sus capacidades, alcance y limitaciones sea de gran interés para mí.

- **Desarrollar una aplicación de libros usados.**

Además, este proyecto se encuentra igualmente relacionado con las investigación y desarrollo de nuevas formas de venta online, que rompan con la tradicional dinámica de tienda online de

artículos nuevos accesible solo desde el ordenador de sobremesa, y que permita explorar las posibilidades que los nuevos dispositivos electrónicos ofrecen al mundo digital. Se trata, por tanto, en último término, de ofrecer un método alternativo de acceso a productos usados, de manera original y llamativa.

1.4 Que se espera con este proyecto

Como potencial profesional de inminente salida al mercado laboral en Ingeniería de Software, el presente TFG gira alrededor de la consolidación de conocimiento de las distintas fases del ciclo de vida del desarrollo del software, como son la Ingeniería de Requisitos, Modelado (funcional y técnico), Arquitectura de Aplicaciones, Desarrollo, Pruebas y despliegue.

En este sentido, a lo largo de mi vida universitaria he adquirido conocimientos de una manera separada de las distintas fases.

Es deseable que este TFG sea el nexo de unión de todas ellas y me ayude a tener la primera visión del desarrollo de un proyecto, a imagen y semejanza de la Ingeniería de Software Empresarial a la que me enfrentaré en un futuro.

Asimismo, es posible que dada la visión Empresarial que me ofrece el Tutor, pueda inclinarme por el uso de tecnologías emergentes no estudiadas por mí, a fin de aumentar el abanico de conocimientos tecnológicos que me ayuden a incorporarme profesionalmente en el mercado.

Como prueba de ello, este TFG se marca como objetivo, la realización de una aplicación empresarial desde cero, partiendo de una base de requisitos no cerrada, un modelado que ofrezca características lo más reales posibles a una tienda online como son: ofrecer un catálogo de artículos (libros en este caso), permitir un registro de usuarios, ofrecer un buscador de libros, permitir un carrito de pedidos, etc. Estas funcionalidades marcarán el diseño de la arquitectura, a la implementación y a las pruebas que se realicen. En todas estas etapas se quiere obtener experiencia y habilidades que me sirvan como base ante el inminente salto al mundo laboral.

1.5 Medios empleados

En este apartado se describen los distintos medios, hardware y software, que se han empleado para el desarrollo del proyecto.

1.5.1 Elementos hardware

Tanto para la implementación como para el desarrollo de la documentación, se han empleado dos equipos informáticos como soporte hardware:

- Ordenador portátil: provisto de un procesador Intel Core i5 3337U, 4GB de memoria RAM, disco duro SSD de 128GB y sistema operativo Windows 8.
- Ordenador portátil Apple MacBook Air 13": con procesador Intel Core i5 de doble núcleo a 1,4 GHz (Turbo Boost de hasta 2,7 GHz) con 3 MB de caché de nivel 3 compartida, 4GB de memoria RAM, disco duro SSD de 256GB y sistema operativo MAC OS X Maverick 10.9.1

1.5.2 Elementos software

En cuanto a las herramientas software utilizadas en el desarrollo del proyecto, mencionar:

- *MAC OS X Maverick 10.9.1.*
- *Windows 8*
- *Microsoft Office 2010*
- *Node.js*
- *MySQL WorkBench 6.0 CE*
- *MongoDB*
- *Sublime Text 3*
- *GitHub*

1.6 Terminología

En este apartado se recoge un glosario en el cual que se explican algunos de los términos utilizados en este documento, con el fin de que el lector comprenda el total contenido del documento.

1.6.1 Acrónimos y abreviaturas

API	- Interfaz de Programación de Aplicaciones
HTML	- HyperText Markup Language (Lenguaje de Marcado de Hipertexto).
IDE	- Integrated Drive Electronics (Unidad Electrónica Integrada).
IMAP	- Protocolo de acceso a mensajes de internos.
IRPF	- Impuesto sobre la Renta de las Personas Físicas.
IVA	- Impuesto sobre el Valor Añadido.
JEE	-Java Enterprise Edition.
LDAP	-Protocolo Ligero de Acceso a Directorios.
JNDI	-Java Naming and Directory Interface.
OSI	-Sistemas de Interconexión Abiertos

- POE -Programación orientada a eventos.
- POP -Protocolo de Oficina de Correo.
- RSA - IBM Rational Software Architect.
- SMTP - Protocolo para la transferencia simple de correo electrónico.
- SQL - Structured Query Language.
- TFG - Trabajo Fin de Grado.

1.6.2 Definiciones

Aplicación web: aquella herramienta que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador.

BSON: formato de intercambio de datos en base de datos MongoDB.

Cliente: es una aplicación informática o un ordenador que consume un servicio remoto en otro ordenador conocido como servidor, normalmente a través de una red de telecomunicaciones.

Hardware: corresponde a todas las partes tangibles de un sistema informático; sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos.

Hilo: es la unidad de procesamiento más pequeña que puede ser planificada por un sistema operativo.

HTML: es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, entre otros.

Git: es un software de control de versiones.

GitHub: es una forja para alojar proyectos utilizando el sistema de control de versiones Git.

Google: buscador web, en él se pueden realizar búsquedas de webs por la World Wide Web a base de un algoritmo exclusivo.

JavaScript: es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico [15].

Mac OS: (Sistema Operativo de Macintosh) es el nombre del sistema operativo creado por Apple para su línea de computadoras Macintosh. Es conocido por haber sido el primer sistema dirigido al gran público en contar con una interfaz gráfica compuesta por la interacción del mouse con ventanas, icono y menús.

Microsoft Office: es una suite de oficina que abarca e interrelaciona aplicaciones de escritorio, servidores y servicios para los sistemas operativos Microsoft Windows y Mac OS X.

Microsoft Windows: es el nombre de una familia de sistemas operativos desarrollados y vendidos por Microsoft.

MongoDB: es un sistema de base de datos NoSQL orientado a documentos.

MySQL: es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

Node.js: es un entorno de programación en la capa del servidor basado en el lenguaje de programación JavaScript, con I/O de datos en una arquitectura orientada a eventos y basado en el motor JavaScript V8.

Intérprete: parte de un compilador, que transforma el texto de entrada en otra estructura, que son más útiles para e posterior análisis.

Plugin: (complemento) es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

Puntero: es una variable que da referencia a una región de memoria.

Serializar: consiste en un proceso de codificación de un objeto en un medio de almacenamiento.

Servidor web: es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente.

Smartphone: o teléfono inteligente es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de computación y conectividad que un teléfono móvil convencional.

Software: al equipamiento lógico o soporte lógico de un sistema informático, comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.

Sistema Operativo: es un programa o conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los restantes.

1.7 Estructura de la memoria

Para ayudar a la fácil lectura de la memoria, a continuación se ofrece un breve resumen de cada capítulo que la comprende.

En el primer apartado: “Capítulo 1. Introducción”, se describen los motivos que han llevado a la realización de este proyecto, así como los objetivos que se desean alcanzar a su finalización. Además se incluye los medios utilizados para llévalo a cabo y un glosario de términos y abreviaturas para facilitar la lectura de la memoria del proyecto.

En el segundo apartado: “Capítulo 2. Estado del arte”, se realiza un análisis del comercio electrónico, realizando un estudio previo de las tecnologías existentes al objetivo del proyecto, un

estudio de las diferentes posibilidades a la hora de realizar un proyecto de estas características y a continuación se proporciona un razonamiento detallado de porqué se ha utilizado una determinada tecnología y que método de desarrollo se ha utilizado para llevar a cabo el proyecto.

En el tercer apartado: “Capítulo 3 Análisis” se definirá lo que se espera del sistema. Para ello se hará una descripción general del sistema y se concretará una lista de requisitos.

En el cuarto apartado: “Capítulo 4 Diseño” se realizará un estudio de los diseños realizados. Y por otro lado el diseño interno, que se dividirá en la creación de diagramas de secuencia, diagrama de clases, de imágenes y para terminar se hablará sobre la persistencia de los datos.

En el quinto apartado: “Capítulo 5 Plan de pruebas” se hará un plan de pruebas acorde con los requisitos definidos en el Capítulo 3.

En el sexto apartado: “Capítulo 6 Gestión del proyecto” se realizará una comparación de la planificación inicial del proyecto y la que realmente se ha llevado a cabo. Para finalizar se realizará un presupuesto de todo el proyecto.

En el séptimo apartado: “Conclusiones” se exponen las conclusiones finales a las que se ha llegado después de la realización del proyecto, haciendo una revisión de los objetivos iniciales y estudiando si se han cumplido debidamente cada uno de ellos. Además se realizará un breve comentario de lo aprendido durante este proyecto de un punto de vista personal. Finalmente se indican una serie de futuros trabajos que se pueden realizar como ampliación al desarrollo de este proyecto.

En octavo y último apartado: “Bibliografía” se indican las páginas y los documentos a las que se ha hecho algún tipo de referencia a lo largo del presente documento.

2. CAPÍTULO 2. ESTADO DEL ARTE

El primer paso que se debe dar en cualquier proyecto es analizar las aplicaciones web existentes en el mercado con el fin de buscar la similitud con nuestra aplicación o detectar carencias o errores para evitar y solucionar dichos problemas en la aplicación a desarrollar. También hay que tener en cuenta el contexto en el cual se va a desarrollar la aplicación.

El segundo paso a realizar es analizar las diferentes plataformas de programación existentes a la hora de desarrollar aplicaciones web que fueron consideradas candidatas a la hora de soportar el desarrollo de la aplicación.

En este capítulo se expone la información obtenida al investigar acerca de los temas que se tratarán a lo largo del desarrollo del proyecto. Para ello, se expone en primer lugar los criterios de búsqueda de las aplicaciones web. Tras esto, un análisis de aplicaciones web relacionadas con los objetivos establecidos. Después un análisis de todas las tecnologías que se podrían usar para resolver este proyecto y para terminar se explica cuál es el entorno de desarrollo elegido y cuáles son sus características principales.

2.1 Criterios de búsqueda

El análisis se va a realizar mediante la búsqueda a través de un buscador web, en este se utilizará el buscador de “google”. Esta búsqueda se basa en aplicaciones web basadas en compra-venta de libros usados. La búsqueda se ha centrado en tiendas online de libros usados ya que es el objetivo principal de nuestro proyecto.

Las herramientas de búsqueda empleados han sido:

- Buscador de Google.

Los términos utilizados para la búsqueda en la web han sido los siguientes:

- “Tiendas online libros usados”
- “Libros usados”
- “Compra-venta libros usados online”
- “aplicaciones e-commerce libros usados”

Estas búsquedas se han realizado en el mes de Junio de 2014.

El siguiente apartado se contemplarán las aplicaciones web que se han aceptado como válidas, las demás aplicaciones encontradas han sido descartadas porque no se ajustaba a los criterios de búsqueda o a los objetivos de la aplicación, ya sea porque no era una aplicación centrada en la compra-venta de libros usados o bien no estaba orientado a los libros.

2.2 Análisis de las aplicaciones existentes en el mercado

2.2.1 AbeBooks



Analizando la situación de las web de compra-venta de libros usados, esta ha sido la mejor posicionada en google, por tanto pasaré a analizar la política que utiliza dicha empresa a la hora de distribuir los libros usados a través de su tienda online [1].

En primer lugar mostraremos un gráfico que describe el funcionamiento general de la tienda y en el cual se pueden ver las distintas partes involucradas:




Ilustración 2: Funcionamiento AbeBooks

Los servicios de AbeBooks están abiertos a todos los libreros profesionales, editoriales, autores o coleccionistas que deseen vender sus libros a todo el mundo.

Es necesario insistir en el hecho que los servicios ofrecidos son especialmente interesantes y rentables a partir de un cierto número de libros (alrededor de 50 libros) a excepción de autores privados o editoriales que deseen promocionar sus libros por otras vías y que por consiguiente, esta web les sirvan como canal de venta.

Por otro lado, todos los miembros de la web deben cumplir un servicio profesional de Atención al Cliente conforme a la ley relativa a la venta por correspondencia y a la Carta de Adhesión de AbeBooks. Como se ha podido comprobar esta web está más enfocada para personas con gran número de libros coleccionados o para librerías que se nutren gracias a la venta de libros antiguos con gran valor. No tanto para un usuario individual que posee libros y quiere sacar algo de dinero después de su lectura, no le sería nada rentable, puesto que requiere de una serie de tasas mensuales, así como una tasa individual por libro, en función de la cuenta que se desee crear.


Pasión por los libros.

[Iniciar sesión](#)
[Su cuenta](#)
[Carrito](#)
[Ayuda](#)

[Búsqueda Avanzada](#)
[Materias](#)
[Libros Antiguos](#)
[Librerías](#)
[Vender Libros](#)

¿Aún no conoce el Boletín de IberLibro? Entérese el primero de ofertas, novedades y mucho más

[Suscribirse](#)

Bienvenido a IberLibro

Encuentre libros puestos a la venta por miles de librerías.

Compre con confianza gracias a nuestra garantía de devolución.

Compra 100% segura. Su privacidad está protegida.

[Saber más](#)

Busque entre millones de libros de miles de librerías

Autor:

Título:

Palabra clave:

¿Qué es? ISBN:

[Buscar](#) [Más opciones de búsqueda](#)

¡Ahorre!

Ofertas y promociones

[Ver descuentos](#)

Gastos de envío gratis

Miles de libros, sin compra mínima

[Saber más](#)

Autor del mes

[Descúbrelo!](#)

Más en IberLibro

- Boletines IberLibro
- Peticiones
- Glosario
- Guía del coleccionista
- Buscar libros por ISBN

[Siganos en Facebook](#)

[Siganos en Twitter](#)

[Siganos en Google+](#)

[Lea nuestro blog](#)

Librerías

- Lista de librerías anticuarias
- HomeBase: Gestione su inventario fácilmente.

Programa de afiliados

- Aumente sus beneficios con IberLibro

Sobre IberLibro.com

IberLibro es un mercado online de alcance mundial con millones de libros nuevos, antiguos, agotados y de ocasión, puestos a la venta por miles de librerías de todo el mundo.

[Más información](#)

Categorías

- Arte y cultura
- Biografías
- Ciencias humanas y sociales
- Ciencias naturales y exactas
- Cocina y gastronomía
- Geografía y cartografía
- Historia
- Infantil y juvenil
- Narrativa y novela
- Religión
- Salud y ejercicio

[Ver catálogo completo](#)

Recomendados

- Bestsellers y novedades
- El exilio de los escritores latinoamericanos
- El Boom latinoamericano
- Novela romántica
- Novela de terror
- Manuscritos iluminados
- 30 bellas cubiertas
- Libros adaptados a películas
- Enciclopedias antiguas
- Música clásica
- Novela negra
- Libros infantiles antiguos
- Libros de viajes insólitos
- El libro más raro
- Bellas obras por menos de 50€
- Libros más caros vendidos

Libros de viajes insólitos

La narrativa de viajes nos sumerge en **aventuras únicas** y nos aporta la particular subjetividad del viajante. llenando nuestra imaginación de matices que construyen entornos completamente nuevos para el lector. En IberLibro queremos compartir con usted nuestro gusto por los buenos libros de viajes y, quién sabe, **quizá descubra su próxima aventura!**

[Ver selección](#)

Libros para el verano

Iberlibro le ofrece una selección de novelas actuales y clásicas, cuentos, libros de humor o **bestsellers**, que refrescaran su verano esté donde esté ¡Entre y prepare su maleta!

[Ver selección](#)

Libros antiguos y de colección

Aquí encontrará un compendio vital para el amante del libro antiguo. Secciones como la de libros firmados, libros raros, facsimiles, librerías anticuarias y nuestro libro destacado del mes.

[Libros antiguos](#)

Librería del mes de Septiembre

La Librería Salambó es una librería anticuaria especializada en libros descatalogados, antiguos y de segunda

Sala de libros usados

Entra en nuestra sala de libros usados y descubre los nuevos temas que le hemos preparado. Libros de consulta, de cocina,

Ilustración 3: Ejemplo web AbeBooks

2.2.2 La casa del libro



Ilustración 4: Logo La casa del libro

La segunda de las opciones que nos aparece en google tras realizar la búsqueda específica ha sido la web de la casa del libro. Dicha web también ofrece un servicio de venta de libros de segunda mano, para que sus usuarios registrados puedan obtener un beneficio de sus libros usados. La web posibilita que sus libros se muestren en un catálogo, accesible por millones de usuarios y el servicio de envío por una pequeña cantidad de dinero.

Como red de intercambio, la web solamente se hace cargo del envío, pero el empaquetado queda de mano del usuario. La web se nutre de obtener el 15% de cada libro vendido por usuario, así como una serie de tasas por tramitación, envío y manipulación, variables en función del número de ventas mensuales por usuario [6].

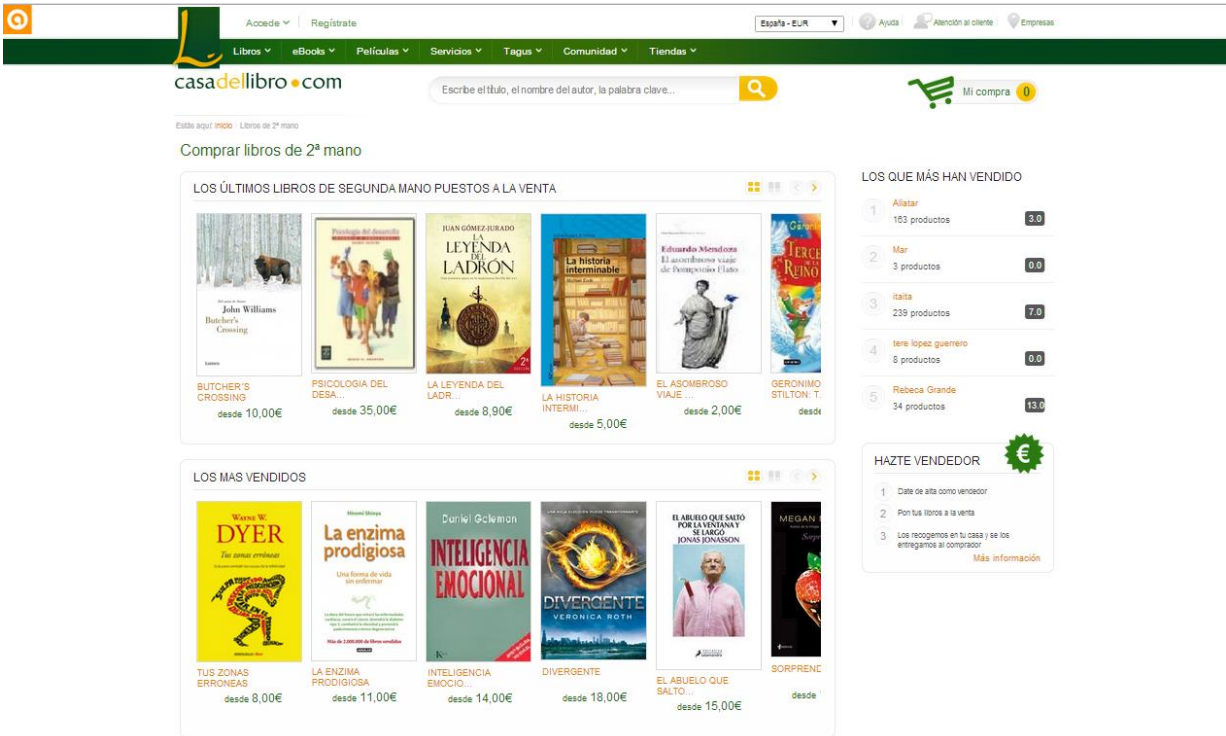


Ilustración 5: Ejemplo web Casa del libro

2.2.3 Amazon



Otra de las opciones que apareció, fue la web de Amazon, en la cual también se pueden vender artículos, y sobre todo libros usados, justo lo que estamos analizando. La política de la empresa es la siguiente, una vez un usuario se ha registrado y ha elegido su tipo de cuenta, vendedor individual o profesional. Podrá subir su catálogo de libros utilizando las herramientas disponibles para ello o por medio de los ficheros de carga para poner en marcha su negocio cuanto antes [3].

En lo referente con las ventas y los envíos, Amazon, posee su propia plataforma de logística, encargada de recibir, preparar y enviar al cliente sus artículos comprados.

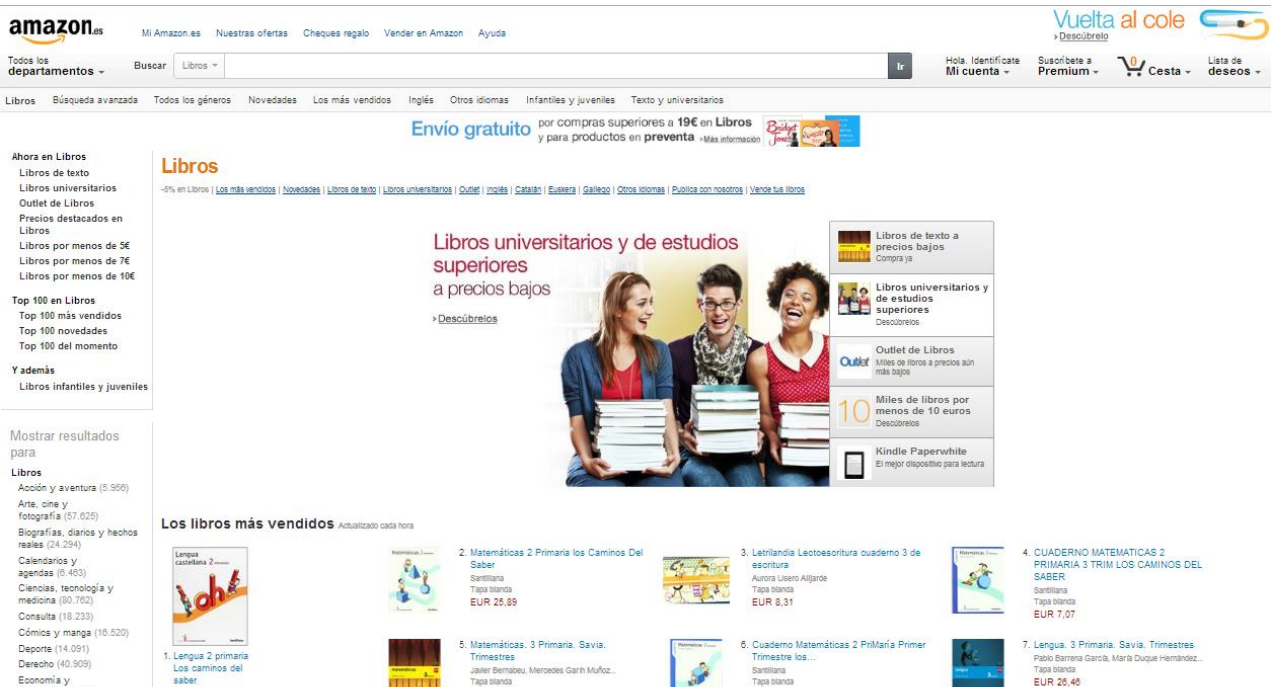


Ilustración 7: Ejemplo web Amazon

2.3 Análisis de alternativas tecnológicas

En este apartado se van a plantear una serie de posibles soluciones capaces de cumplir con lo que se requiere a la hora de desarrollar una tienda online. Para, seguidamente, analizarlas, presentar sus ventajas y elegir la más adecuada.

Debido al carácter implícito cliente-servidor del sistema, es necesario que al presentar las alternativas se distinga entre las tecnologías del servidor y las tecnologías del cliente.

Pese a estar dividido en cliente servidor, para realizar el análisis hay que pensar en las alternativas como un conjunto inseparable, ya que la compatibilidad entre ellas será uno de los rasgos que puedan llevar a la elección o descarte de una.

También se debe destacar que nuestro país es uno de los que mayor desarrollo está experimentando en el avance del software libre, gracias, entre otras cosas, al apoyo de las administraciones públicas, tanto locales como autonómicas.

2.3.1 Alternativa 1 – Desarrollo aplicación web con JEE

El lenguaje de programación usado en JEE es Java. La plataforma JEE está orientada a la construcción de aplicaciones distribuidas y multicapa. Para lograrlo, se ofrecen diferentes servicios basados en distribución de objetos, gestión de transacciones, seguridad, y gestión de recursos [17].

Al proveer una infraestructura empresarial, varias características de JEE disminuyen el esfuerzo de los desarrolladores en la codificación. Así, los desarrolladores sólo se enfocan en la implementación de los componentes distribuidos que forman parte de la lógica del negocio [18].

Sobre el lenguaje Java

- Orientado a objetos.
- Basado en componentes: Java tiene diferentes componentes que permiten construir aplicaciones empresariales acelerando el desarrollo.
- Arquitectura independiente de plataforma.
- Recolector de basura.
- Libre de punteros.
- Soporte para hebras.
- Serialización de objetos para integración con otras aplicaciones Java.
- Soportado en algunos dispositivos móviles.
- Varias API para servicios como mensajería, LDAP, conectividad con bases de datos, etc.

Características de la plataforma

- Servlets y JavaServerPages para la construcción de aplicaciones web.
- Enterprise JavaBeans que permiten escribir código sin detalles de implementación de persistencia, transacciones, comunicación remota y otros servicios.
- Java Transaction API que permite tener transacciones por aplicación, incluso en aquellas que son distribuidas.
- Java MessageService que permite utilizar una API completa para compartir mensajes entre una o diferentes aplicaciones.
- Java Naming and Directory Interface (JNDI) que provee soporte para localización de servicios.
- JavaMail que provee soporte SMTP, IMAP y POP.

Requisitos para el uso

Aunque no es requerido un entorno para el desarrollador, existen entornos que facilitan la tarea de codificación. Por otro lado, el entorno de ejecución funciona en cualquier plataforma.

Entorno para el desarrollador

Cabe destacar tres entornos de desarrollo recomendados para construir aplicaciones con JEE. Uno de ellos es NetBeans [24] el oficial de Sun Microsystems. Otro es Eclipse [10], que es un entorno de desarrollo muy flexible y tiene un buen repositorio de plugins para extender sus funcionalidades. Por último destacar la herramienta de IBM Rational Software Architect [27], basado en la infraestructura de software de código abierto Eclipse y se puede ampliar con varios plug-ins de Eclipse.

Entorno de ejecución

La plataforma J2EE puede ser instalada en cualquier sistema operativo y no se debe comprar licencias para su uso.

Para su ejecución se requiere instalar un contenedor de aplicaciones, como Apache Tomcat [4], GlassFish [13] o IBM WebSphereApplication Server [33].

En la parte del servidor existen una gran abanico de posibilidades para alojar la aplicación web y poderla ejecutar sin problema. Más abajo se mencionarán alguno de los servidores que se han utilizado en conjunto con la plataforma de programación (JEE) y el entorno de desarrollo (IBM RSA):

- Apache Tomcat
- Glassfish
- IBM WebSphere Application Server

2.3.2 Alternativa 2 - Desarrollo aplicación web con Node.js



Ilustración 8: Logo Node.js

Node.js es una plataforma construida encima del entorno de ejecución JavaScript de Google Chrome, el cual permite fácilmente construir aplicaciones de red. Usa un modelo de E/S no bloqueante dirigido por eventos que lo hace ligero y eficiente, perfecto para aplicaciones “data-intensive” en tiempo real [26].

Sobre Node

Es una tecnología asíncrona lo que nos ofrece una mayor respuesta en cuanto a peticiones de los usuarios, como se puede mostrar en la siguiente imagen que explica una petición en diferentes arquitecturas:

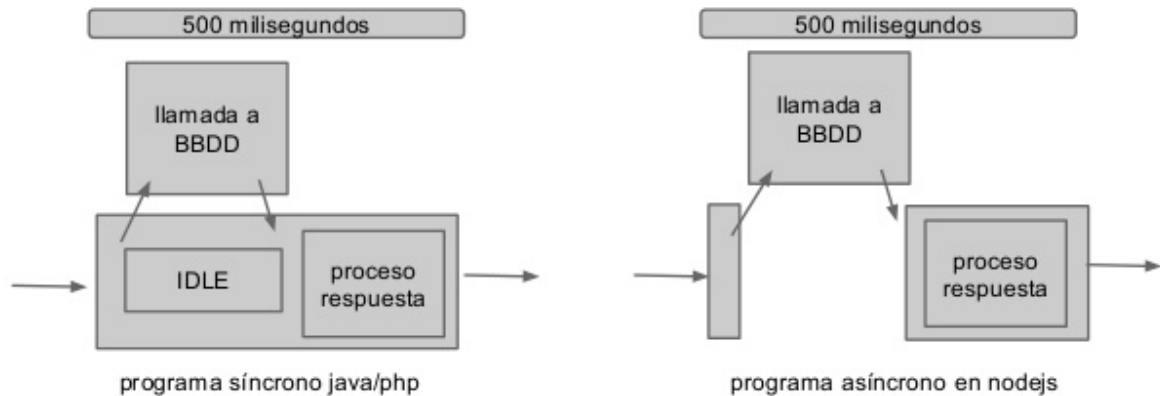


Ilustración 9: Comparativa Java vs Node.js

Características de Node.js

A continuación se describen las características [25] de Node.js:

- Maneja mucho mejor la concurrencia masiva ya que es asíncrono y es más rápido que otras.
- Utiliza el lenguaje de programación JavaScript. Todo se ejecuta en el servidor, olvidando así si el código es compatible con el cliente.
- Programación asíncrona. Se crean diferentes hilos, con diferentes procesos que llevarán un tiempo en ejecutarse, para poder ejecutar todo a la vez. Pudiendo especificar que fragmento de código (callbacks) se ejecute al final de cada uno de esos procesos largos.
- EL uso intenso de callbacks produce un efecto de funciones metidas en otras funciones. Produciendo un código menos sencillo de entender visualmente y difícil de mantener.
- Programación orientada a eventos(POE)

Requisitos para su uso

Para poder utilizar Node.js es necesario obtenerlo pre-compilado para una distribución Windows o Mac y en el caso de otras distribuciones Linux es necesario de su compilación que está basada en C++.

Existe la posibilidad de instalarlo en las diferentes arquitecturas y sistemas operativos. Por tanto para poder utilizarlo basta con visitar su página web y descargarse la versión estable del mismo.

Entorno para el desarrollador

Existen numeroso entorno de desarrollo, así como IDE que permiten utilizar Node.js, pero realmente no se necesita de ninguna herramienta potente para editar el código de esta plataforma, por lo que se ha decidido utilizar el editor de texto " Sublime Text 3" que se adapta a un gran número de lenguajes y que además ofrece un gran número de características que hace que se adapte muy bien a las numerosas llaves, llamadas y cierres de etiquetas que tiene Node.js, ofreciendo algo muy importante también, una cómoda navegación entre los directorios del proyecto.

Entorno de ejecución

Por otro lado, la gran diferencia con JEE, en que la aplicación web no se ejecuta en un servidor de aplicaciones existente, como Tomcat, etc., el propio Node.js incorpora accesibilidad nativa a los socket del sistema operativo, permitiendo crear conexiones en las capas inferiores de la tabla OSI, dando la posibilidad de crear un servicio totalmente ajustable a los requisitos de la aplicación.

Por tanto, la utilización de Node.js se antoja algo más complicada que JEE, ya que se trata de una nueva tecnología y con herramientas y formas de programar diferentes, aunque al existir gran número de documentación por la red, no hace que este hecho sea muy complicado. Por tanto la parte más diferenciada con la alternativa anterior es sobre todo en lo relacionado con el entorno de ejecución, ya que habría que construir un entorno propio.

2.4 Análisis de metodologías para realizar el proyecto

2.4.1 Metodologías ágiles

Las Metodologías Ágiles constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos que las metodologías convencionales (ISO-9000, CMM, etc.) debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

Las metodologías de desarrollo ágiles son adaptativas en lugar de predictivas. Los métodos de ingeniería tienden a intentar planear una parte grande del proceso del software en gran detalle para un plazo largo de tiempo, esto funciona bien hasta que las cosas cambian. Así que su naturaleza es resistirse al cambio. Para estas metodologías, no obstante, el cambio es bienvenido.

Intentan ser procesos que se adaptan y crecen en el cambio, incluso al punto de cambiarse ellos mismos. Son orientadas a la gente y no orientados al proceso. Y su meta es definir un proceso que funcionará bien con cualquiera que lo use [21].

2.4.2 The Agile Alliance

En febrero de 2001, tras una reunión celebrada en Utah (EE.UU.), nace el término ágil aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras esta reunión se creó “The Agile Alliance”, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos [2]. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía ágil, es este manifiesto se valora:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- Desarrollar software que funciona más que conseguir una buena documentación. La regla a seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental.
- La colaboración con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

Dentro de las metodologías ágiles una de las características principales es el uso de historias de usuario. Son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

2.4.3 Scrum

Por último se va a hacer referencia a la metodología ágil utilizada como base para la realización de este proyecto que es Scrum¹⁶ y cómo se ha adaptado a nuestro entorno.

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión (ROI) para su empresa.

Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación.

Con Scrum, el cliente se entusiasma y se compromete con el proyecto dado que lo ve crecer iteración a iteración. Asimismo le permite en cualquier momento realinear el software con los objetivos de negocio, ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración.

Esta metódica de trabajo promueve la innovación, motivación y compromiso del equipo que forma parte del proyecto, por lo que los profesionales encuentran un ámbito propicio para desarrollar sus capacidades.

Los principales beneficios que proporciona la utilización de Scrum son los siguientes:

- **Cumplimiento de expectativas:** El cliente establece sus expectativas indicando el valor que le aporta cada requisito/historia del proyecto, el equipo los estima y con esta información establece su prioridad. De manera regular, en las demos de Sprint se comprueba que efectivamente los requisitos se han cumplido y se transmite feedback (retroalimentación) al equipo.
- **Flexibilidad a cambios:** Alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado.
- **Reducción del Time to Market:** El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo.
- **Mayor calidad del software:** La metódica de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.
- **Mayor productividad:** Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse.
- **Maximiza el ROI:** Producción de software únicamente con las prestaciones que aportan mayor valor de negocio gracias a la priorización por retorno de inversión.
- **Predicciones de tiempos:** Mediante esta metodología se conoce la velocidad media del equipo por sprint (los llamados puntos historia), con lo que consecuentemente, es posible estimar fácilmente para cuando se dispondrá de una determinada funcionalidad que todavía está como trabajo pendiente.
- **Reducción de riesgos:** El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada.

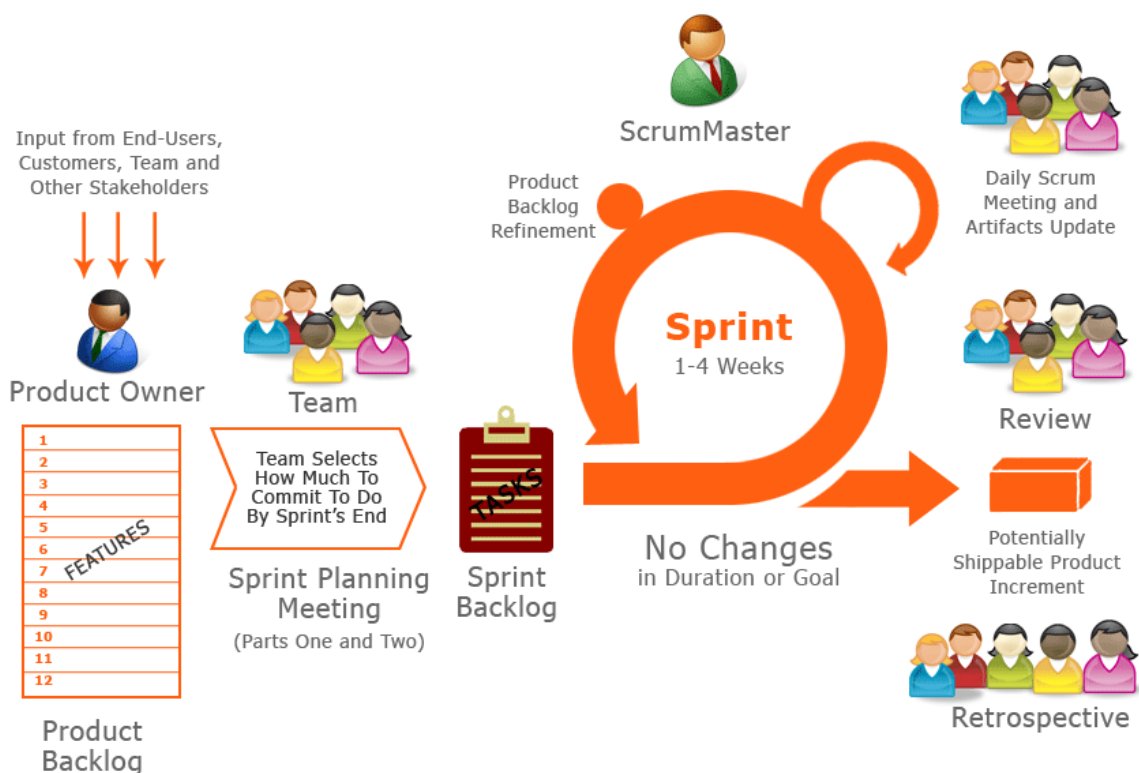


Ilustración 10. Metodología SCRUM.

2.4.4 Metodología tradicional.

Las Metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada.

Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

Entre las metodologías tradicionales se va a proceder a describir las siguientes:

2.4.4.1 RUP (Rational Unified Procces)

El Rational Unified Process (RUP) es un proceso de desarrollo de software elaborado por la empresa Rational Software, actualmente propiedad de IBM. Junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos [28].

Este proceso de desarrollo de software no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

El RUP está basado en los siguientes principios clave:

- **Adaptar el proceso:** el proceso deberá adaptarse a las necesidades del cliente ya que es muy importante interactuar con él.
- **Equilibrar prioridades:** los requisitos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un equilibrio que satisfaga los deseos de todos.
- **Demostrar valor iterativamente:** los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.
- **Colaboración entre equipos:** El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requisitos, desarrollo, evaluaciones, planes y resultados.
- **Elevar el nivel de abstracción:** un alto nivel de abstracción también permite discusiones sobre diversos niveles y soluciones arquitectónicas. Éstas se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con el lenguaje UML.
- **Enfocarse en la calidad:** el control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

Principales características.

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

Fases internas de RUP

RUP comprende dos aspectos importantes por los cuales se establecen las disciplinas:

1. **'Proceso':** Las etapas de esta sección son:

- Modelado de negocio

- Requisitos
- Análisis y Diseño
- Implementación
- Pruebas
- Despliegue

2. **Soporte:** En esta parte nos encontramos con las siguientes etapas:

- Gestión del cambio y configuraciones
- Gestión del proyecto
- Entorno

La estructura dinámica de RUP permite que éste sea un proceso de desarrollo fundamentalmente iterativo, y en esta parte se ven inmersas las cuatro fases descritas anteriormente:

1. **Fase de Inicio:** Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.
2. **Fase de elaboración:** En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.
3. **Fase de Desarrollo:** El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.
4. **Fase de Transición:** El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

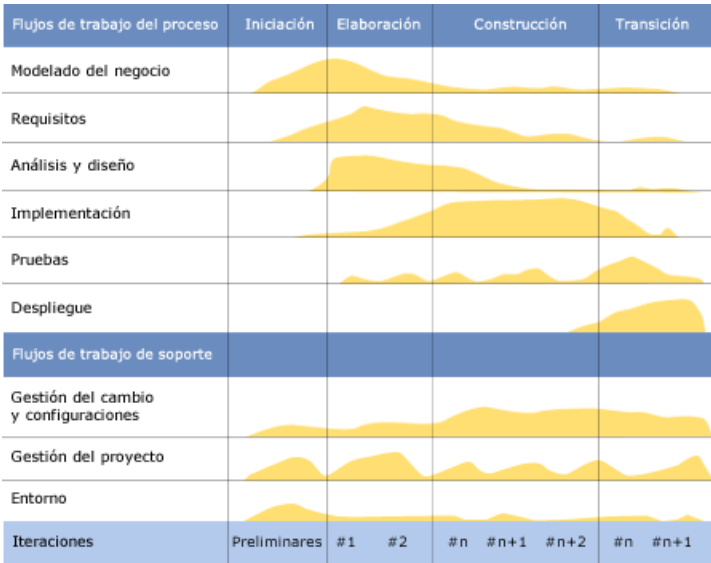


Ilustración 11. Resumen estructura dinámica RUP

2.4.4.2 Modelo en cascada.

Es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase [9].

Fases Método en cascada.

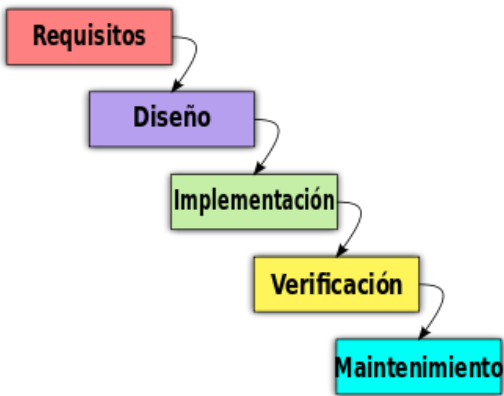


Ilustración 12. Fases del método en cascada.

- **Análisis de requisitos**

En esta fase se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir.

- **Diseño del Sistema**

Descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo.

Es conveniente distinguir entre diseño de alto nivel o arquitectónico y diseño detallado. El primero de ellos tiene como objetivo definir la estructura de la solución. El segundo define los algoritmos empleados y la organización del código para comenzar la implementación.

- **Diseño del Programa**

Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber qué herramientas usar en la etapa de Codificación

- **Codificación**

Es la fase en donde se implementa el código fuente, haciendo uso de prototipos así como de pruebas y ensayos para corregir errores.

- **Pruebas**

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente y que cumple con los requisitos, antes de ser entregado al usuario final.

- **Verificación**

Es la fase en donde el usuario final ejecuta el sistema, para ello el o los programadores ya realizaron exhaustivas pruebas para comprobar que el sistema no falle.

- **Mantenimiento**

Una de las etapas más críticas, ya que se destina un 75% de los recursos, es el mantenimiento del Software ya que al utilizarlo como usuario final puede ser que no cumpla con todas nuestras expectativas.

Ventajas método en cascada

- Realiza un buen funcionamiento en equipos débiles y productos maduros, por lo que se requiere de menos capital y herramientas para hacerlo funcionar de manera óptima.
- Es un modelo fácil de implementar y entender.
- Está orientado a documentos.
- Es un modelo conocido y utilizado con frecuencia.
- Promueve una metodología de trabajo efectiva: Definir antes que diseñar, diseñar antes que codificar.

2.4.4.3 Comparativa entre metodologías ágiles y tradicionales

Para finalizar este apartado, se va a realizar una comparativa entre las metodologías analizadas, en la siguiente tabla se muestra los puntos fuertes de cada una:

Metodología Ágil	Metodología No Ágil (Tradicional)
Pocos artefactos	Más artefactos
Pocos roles	Más roles
No existe un contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes
Menos énfasis en la arquitectura	La arquitectura es esencial

Tabla 1. Diferencias entre Metodología ágil y metodología tradicional.

2.5 Conclusiones

➤ En lo referente al mercado de las aplicaciones de e-commerce

Tras realizar un análisis de diferentes posibilidades que se pueden encontrar en la web el método que utilizan la gran mayoría de las web que ofrecen la posibilidad de vender libros de segunda mano, les cobran una serie de tarifas mensuales, les ofrecen un catálogo donde mostrar sus libros y les ofrecen una serie de servicios de cara al envío de los productos a sus compradores. Además de

ofrecen una interfaz dinámica donde el usuario puede acceder a todo tipo de datos acerca del libro que esté buscando.

Por todos estos motivos se ha pensado en hacer una aplicación web en la cual el usuario pueda subir sus libros para mostrarlos en su catálogo. Dicha web será únicamente una plataforma de enlace entre usuarios, ya que no cobrará ningún tipo de tarifa a los usuarios de cara al uso de su web. Solamente será una comunidad en la que los usuarios podrán vender a otros usuarios registrados de la web y comprar también de otros usuarios.

En la web que se va a llevar a cabo contendrá una mezcla de todos los puntos fuertes de todas las tiendas online que se han visitado e intentando alejarse de sus puntos débiles.

Después de explicar las características de cada una de las web mencionadas, se va a proceder a realizar un análisis comparativo entre ellas teniendo en cuenta algunos puntos como:

- La facilidad de captar la atención de los usuarios, es decir, si el usuario es capaz de estar visualizando la web hasta que llegue a su objetivo.
- La sencillez de la aplicación, es decir, que no contenga gran cantidad de texto y que pueda resultar relativamente sencillo completar cualquier tipo de servicio.
- La originalidad en cuanto a imágenes y textos, es decir, si le impresiona al usuario las imágenes que visualiza o si le es fácil leer los contenidos de texto.
- El coste por utilizar la aplicación web. En este caso nuestra web será de registro gratuito y ofrecerá el servicio de compra-venta de forma gratuita, no así como el envío que correrá por parte del usuario.

El resultado de la comparación se puede consultar en la Tabla que se muestra a continuación:

Objetivos	Captar atención	Sencillo	Original	Precio
Web				
AbeBooks	Bajo	Medio	Bajo	Medio
La casa del libro	Medio	Medio	Alto	Bajo
Amazon	Alto	Bajo	Alto	Alto

Tabla 2: Comparativa características web.

➤ En lo referente a las tecnologías analizadas

Tal y como se ha visto en estas dos alternativas anteriores, descritas en el apartado “2.3. Análisis de alternativas tecnológicas”, existen varias formas para desarrollar aplicaciones web, cada una de ellas con sus ventajas y sus inconvenientes, pero en general ambas podrían ser utilizadas a la hora de desarrollar el objetivo de este proyecto, la tienda online de libros usados multi-dispositivo. Por lo que

ahora se describirá por qué una ha sido la elegida para la realización del proyecto en detrimento de la otra.

Como ya se mencionó en el apartado “1.4. *Que se espera de este proyecto*”, el desarrollo de este TFG busca un aprendizaje y tras realizar un análisis de las alternativas, se ha podido observar que la primera de ellas, me resulta conocida, ya que sería utilizar numerosas de las herramientas vistas a lo largo del curso en una asignatura de la carrera, Tecnologías Informáticas para la Web. Mientras que la otra alternativa consta, de la utilización de nuevas tecnologías emergentes en el desarrollo de aplicaciones de e-commerce.

Por tanto por el hecho de querer conocer e investigar acerca de cómo se comportan estos nuevos avances, se ha decidido que las tecnologías propuestas en la segunda alternativa, sean las que se lleven a cabo a la hora de desarrollar este TFG.

➤ En lo referente a la metodología

En el ámbito de las metodologías, este TFG quiere experimentar como afecta el uso de una determinada metodología u otra a la hora de realizar el proceso de desarrollo de un software.

En la actualidad, un alto porcentaje de empresas dedicadas al desarrollo de aplicaciones web, eligen una metodología ágil, a la hora de realizar sus procesos de desarrollo de software. Produciendo un cambio, no solo en la manera de desarrollar, también organizativo y de manera de pensar.

Pero todavía muchas otras siguen utilizando las metodologías tradicionales, las cuales sufren problemas durante su desarrollo, ya que sufren problemas cambiantes. Este se debe a que los usuarios tienden a saber qué quieren una vez visto el proceso realizado.

Por ende, nos encontramos con este TFG, el cual quiere experimentar en su desarrollo todos los problemas o beneficios del uso del método de desarrollo en cascada, pero con una variante, dejando los requisitos de usuario abiertos, es decir, que se irán modificando a lo largo del desarrollo. Se quiere comprobar lo duro y difícil que es producir un producto software por esta vía, al igual que sucede en las empresas.

Esta elección me servirá para saber cómo actuar ante los constantes cambios que se producen en el desarrollo de un producto software en mi futura carrera profesional.

2.6 Solución propuesta

2.6.1 Entorno tecnológico seleccionado

Este capítulo describe las diferentes herramientas y tecnologías usadas para el apoyo del desarrollo, o para la creación en sí, de este proyecto. Para facilitar al lector la lectura del capítulo se ha separado en dos secciones principales: las herramientas usadas en el cliente y las herramientas utilizadas en la parte del servidor.

2.6.1.1 Herramientas y tecnologías del cliente

Vista de la aplicación

Las herramientas del cliente están orientadas al desarrollo de la aplicación web, hay que tener en cuenta que la tecnología usada va prefijada con la elección del uso la aplicación. Como en la actualidad la gran mayoría de las personas no solo utilizan su ordenador para realizar consultas por Internet, hemos desarrollado la aplicación utilizando una herramienta que nos ofrece la posibilidad de cambiar el tipo de vista en tiempo real entre los diferentes dispositivos, en función del tamaño de su pantalla para ofrecer una presentación más adaptable e intuitiva, evitando que no afecte a la visualización de la interfaz en cada una de ellas.

Por tanto a la hora de realizar la interfaz del usuario se necesita que satisfaga los objetivos comentado en el apartado anterior.

Por lo que se va a utilizar el framework de hojas de estilo y JavaScript, Bootstrap, el cual facilita a la hora de crear interfaces adaptables a cualquier tipo de tamaño de pantalla de dispositivo móvil, convirtiendo la interfaz en una web dinámica en función del dispositivo que solicite su visualización.



Ilustración 13: Logo de Bootstrap.

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo [5].

El beneficio de usar “responsive design” en un sitio web, es principalmente que el sitio web se adapta automáticamente al dispositivo desde donde se acceda. Cada día esta tecnología está siendo utilizada con mayor frecuencia, por lo que un gran número de empresas solicitan como requisito a sus futuros empleados. Entendiendo por ello que es una tecnología en auge.

Los diseños creados con Bootstrap son simples, limpios e intuitivos, esto les da legibilidad a la página, a la par que ofrece que ofrece simplicidad a la hora de su desarrollo.

Bootstrap en su desarrollo e implementación utiliza los siguientes lenguajes de programación:



Ilustración 14: Lenguajes programación Bootstrap.

Persistencia de datos

En lo referente a la persistencia de datos, se va a hacer uso de dos tipos de base de datos, SQL y NonSQL. De ésta última tecnología de almacenamiento de datos, existe poco tratamiento didáctico, dada a su reciente implementación en el mercado.

La primera de ellas, gestionará todo lo referente a los datos de los usuarios registrados en la web, así como los libros subidos y los diferentes datos de cada libro.

Por consiguiente, la última, almacenará tanto la sesión del usuario, como los pedidos que realice.

Descripción de las diferentes BBDD utilizadas



Ilustración 15: Logo MySQL

MySQL es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos propietarios pueden comprar a la empresa una licencia más permisiva que les permita ese uso.

MySQL AB es una compañía de software fundada en 1995, creador del sistema administrador de bases de datos relacionales MySQL, y una de las más grandes empresas de software libre del mundo. Sin embargo en 2008 fu absorbida por Sun Microsystems y esta a su vez por Oracle Corporation, lo que está llevando a una pérdida de su filosofía de GNU.

La gran cantidad de aplicaciones de MySQL hace que sea muy utilizado en aplicaciones web como MediaWiki o Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy

rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones [23].



Ilustración 16: Logo MongoDB

Es un sistema de base de datos No SQL orientado a objetos, desarrollado bajo el concepto de código abierto.

MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL. En vez de guardar los datos en tablas como se hace en las base de datos relacionales, guarda estructuras de datos en documentos tipo BSON con un esquema dinámico, haciendo que la integración de datos sea más fácil y rápida [22].

Características de MongoDB

- Soporta la búsqueda por campos, consultas de rangos y expresiones regulares. Las consultas pueden devolver un campo específico del documento pero también puede ser una función JavaScript definida por el usuario.
- Cualquier campo en un documento de MongoDB puede ser indexado
- Permite el tipo de replicación maestro-esclavo. El maestro puede ejecutar comandos de lectura y escritura. El esclavo puede copiar los datos del maestro y sólo se puede usar para lectura o para copia de seguridad, pero no se pueden realizar escrituras. El esclavo tiene la habilidad de poder elegir un nuevo maestro en caso de que se caiga el servicio con el maestro actual.
- Puede ser utilizado con un sistema de archivos, tomando la ventaja de la capacidad que tiene MongoDB para el balanceo de carga y la replicación de datos utilizando múltiples servidores para el almacenamiento de archivos.
- Tiene la capacidad de realizar consultas utilizando JavaScript, haciendo que estas sean enviadas directamente a la base de datos para ser ejecutadas.

2.6.1.2 Herramientas y tecnologías del servidor

Como se mencionó al final de la descripción de las alternativas y durante la elección de la misma, Node.js, requiere de la creación de un servidor propio con el lenguaje nativo para ejecutar la aplicación y que funcione correctamente.

Por tanto para el tema del servidor, no se va a utilizar ninguno prediseñado, sino que creará uno con las herramientas y funciones que trae consigo Node.js, el funcionamiento y la creación del mismo se explicará en el *Capítulo 4 “Diseño del sistema”*.

2.6.1.2.1 Requerimientos hardware

A la hora de desplegar el servidor, se necesita de una máquina con las siguientes características hardware:

- Como mínimo un procesador Intel Core i5 o superior.
- Mínimo 4 GB de memoria RAM DDR3

Con estas características se garantiza el correcto funcionamiento del servidor, para que pueda proveer al cliente de suministros de forma correcta.

2.6.1.3 Herramienta de desarrollo

Como se dijo en el análisis de la alternativa 2, Sublime Text 3 ofrece una gran facilidad para el trabajo con este lenguaje, por tanto para el desarrollo de este proyecto se utilizará este editor de texto. A continuación se detallaran las funcionalidades que ofrece la herramienta elegida



Ilustración 17: Logo de Sublime Text 3

Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo “vi” de Linux llamado Vintagemode [30].

Tiene un gran número de características:

- Mini mapa: consiste en una pre visualización de la estructura del código, es muy útil para desplazarse por el archivo cuando se conoce bien la estructura de este.
- Multiselección: Hace una selección múltiple de un término por diferentes partes del archivo.

- Multi Cursor: Crea cursores con los que podemos escribir texto de forma arbitraria en diferentes posiciones del archivo.
- MultiLayout: Trae siete configuraciones de plantilla podemos elegir editar en una sola ventana o hacer una división de hasta cuatro ventanas verticales o cuatro ventanas en cuadrícula.
- Soporte nativo para infinidad de lenguajes: Soporta de forma nativa 43 lenguajes de programación y texto plano.
- SyntaxHighlight configurable: El remarcado de sintaxis es completamente configurable a través de archivos de configuración del usuario.
- Búsqueda Dinámica: Se puede hacer búsqueda de expresiones regulares o por archivos, proyectos, directorios, una conjunción de ellos o todo a la vez.
- Auto completado y marcado de llaves: Se puede ir a la llave que cierra o abre un bloque de una forma sencilla.
- Soporte de Snippets y Plugins: Los snippets son similares a las macros o los bundles además de la existencia de multitud de plugins.
- Configuración total de Keybindings: Todas las teclas pueden ser sobrescritas a nuestro gusto.
- Acceso rápido a línea o archivo: Se puede abrir un archivo utilizando el conjunto de teclas Cmd+P en Mac OS X o Ctrl+P en Windows y Linux y escribiendo el nombre del mismo o navegando por una lista. También se puede ir a una línea utilizando los dos puntos ":" y el número de línea.
- Paleta de Comandos: Un intérprete de Python diseñado solo para el programa con el cual se puede realizar infinidad de tareas.
- Coloreado y envoltura de sintaxis: Si se escribe en un lenguaje de programación o marcado, resalta las expresiones propias de la sintaxis de ese lenguaje para facilitar su lectura.
- Pestañas: Se pueden abrir varios documentos y organizarlos en pestañas.
- Resaltado de paréntesis e indentación: cuando el usuario coloca el cursor en un paréntesis, corchete o llave, resalta esta y el paréntesis, corchete o llave de cierre o apertura correspondiente.

2.6.1.4 Repositorio del proyecto



Ilustración 18: Logo de GitHub.

A la hora de trabajar desde varias máquinas sobre un mismo proyecto, resulta difícil la tarea de mantener un código legible y actualizado en ambas máquinas. Para evitar que se transporte todo

sobre un soporte físico, se ha hecho uso de un repositorio que se encargue de almacenar los cambios generados en el código, además de avisar de los cambios realizados desde las diferentes máquinas conectadas al repositorio. Para realizar esta función se ha elegido GitHub, herramienta usada para alojar proyectos, el cual permite el sistema de control de versiones de Git [12].

Características de GitHub

- Posee una Wiki para el proyecto, para intercambiar y documentar el proyecto.
- Existe un gráfico detallado, de cómo cada colaborador está trabajando en el proyecto.
- Se puede tener alojada una página web para la presentación del repositorio, que se tenga alojado.

2.6.1.5 Metodología de desarrollo elegida

Como se ha analizado en el “*apartado 2.4. Método para realizar el proyecto*”, existen varias metodologías de trabajo a la hora de realizar el desarrollo de un proyecto software.

Normalmente a la hora de desarrollar aplicaciones web de *e-commerce*, se suelen utilizar metodologías ágiles por todo lo que ofrecen de ajustarse a los continuos cambios sin verse afectadas ninguna parte, ni tan siquiera tener que rediseñar de nuevo.

Aunque está en auge el uso de estas metodologías, existe un gran número de empresas que utilizan las metodologías no ágiles, las cuales se planifica todo desde el principio y se comienza etapa por etapa, aun sabiendo que a lo largo del proyecto aparecerán cambios que obligaran a volver a planificar y a rediseñar todo lo anterior.

Se ha elegido esta metodología para enfrentarme a un caso en el que la metodología no se aplica al 100%, al no disponer de los requisitos cerrados, lo ideal sería utilizar una metodología ágil. Aún en esta casuística, muchas compañías optan por un modelo en cascada y ello conlleva constantes replanificaciones ante los cambios (no es posible seguir un plan, cascada utiliza un modelo productivo, frente al modelo adaptativo de las ágiles). Mi objetivo es enfrentarme a esta problemática a lo largo del desarrollo de este TFG.

3. CAPÍTULO 3. ANÁLISIS DE REQUISITOS

Una vez conocido el entorno tecnológico y el estado de la cuestión, se definirá clara y detalladamente lo que se espera del sistema. Es decir, las funcionalidades requeridas. Para ello se dará una descripción general del sistema como idea aproximada. Se concretará luego una lista de requisitos de usuario. A partir de los requisitos se generarán los casos de uso. Por último se establecerá, en base a lo anterior, la lista de requisitos software del sistema.

3.1 Descripción general

Como se ha comentado, el objetivo de este proyecto es implementar una tienda online para que los usuarios puedan realizar la compra-venta de libros usados utilizando como eje central del mismo la plataforma Node.js. Para que puedan realizar el proceso de venta de libros usados, los usuarios deben estar registrados en la web, así como si quieren comprar algún libro de otro usuario. Además tanto los usuarios registrados como los no registrados podrán enviar un mensaje de consulta a los administradores de la web.

Las diferentes secciones de la web poseen contenido en función del rol del usuario, para otorgar privilegios a los usuarios registrados.

Las funcionalidades que se permiten en la aplicación son:

- Registrarse en la aplicación
- Poder enviar mensaje de contacto
- Poder visualizar el catálogo de libros
- Poder visualizar detalles de cada libro individualmente
- Poder visualizar menú personal de usuario y una vez dentro del menú personal:
 - Poder subir un libro al catálogo de libros disponibles
 - Poder actualizar los datos personales
 - Poder visualizar los datos personales
 - Poder ver histórico de libros subidos
 - Poder ver los pedidos realizados
 - Poder ver el carrito del pedido actual.
- Poder buscar un libro por su título
- Poder ordenar los libros mostrados en el catálogo
- Poder acceder al menú personal
- Poder añadir un libro al carro de la compra
- Poder finalizar el pedido de libros
- El usuario administrador podrá gestionar usuarios
- El usuario administrador podrá gestionar libros

3.2 Requisitos de usuario

Los requisitos de usuario son los primeros a definir en el análisis y deben ser descritos por el cliente, que en este caso concreto es el propio alumno desarrollador de la aplicación. El patrón de especificación de requisitos es el siguiente:

Identificador			
Título			
Descripción			
Fuente		Necesidad	
Prioridad		Claridad	
Estabilidad		Verificabilidad	

Tabla 3. Plantilla requisitos de usuario

Detalle de la tabla:

- Identificador: Código único que identifica al requisito. La nomenclatura se basa en el formato RUX-YY, donde X es la inicial del tipo de requisito (Restricción o Capacidad) y donde YY es un número correlativo incremental.
- Título: Título descriptivo de cada requisito.
- Descripción: Breve descripción del requisito. Debe ser coherente y evitar la ambigüedad
- Fuente: especifica el origen del requisito.
- Necesidad: especifica el grado de importancia que tiene este requisito para el proyecto. Los valores que puede tomar este campo son: esencial, deseable u opcional.
- Prioridad: especifica la urgencia con la que el requisito debe cumplirse. Los valores que puede tomar este campo son: alta, media o baja.
- Claridad: especifica si el requisito es entendible por sí solo y de manera correcta. Los valores que puede tomar este campo son: alta, media o baja.
- Estabilidad: especifica si el requisito puede variar a lo largo de la vida del proyecto. Los valores que puede tomar este campo son: estable o inestable.
- Verificabilidad: especifica si el cumplimiento del requisito puede comprobarse. Los valores que puede tomar este campo son: alta, media o baja.

3.2.1 Requisitos de capacidad

Ahora se mostraran todos los requisitos de capacidad, que recogen lo que necesita un usuario para lograr su objetivo:

RUC-01			
Titulo	Registro de usuario		
Descripción	El usuario podrá registrarse en la web utilizando el formulario de registro.		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 4. RUC-01. Registro de usuario.

RUC-02			
Titulo	Acceso a la web		
Descripción	El usuario accederá a la web utilizando sus credenciales.		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 5. RUC-02: Acceso a la web.

RUC-03			
Titulo	Acceso menú personal		
Descripción	<div>El usuario conectado a la web podrá acceder a su menú personal permitiéndole:</div> <ul style="list-style-type: none">• Tener acceso a sus datos personales• Modificar sus datos personales• Subir libros al catálogo de libros disponibles• Tener acceso a sus libros subidos• Tener acceso a su historial de pedidos• Tener acceso a su carro de pedido actual.		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 6. RUC-03. Acceso menú personal.

RUC-04			
Titulo	Buscar libro		
Descripción	<div>El usuario podrá realizar una búsqueda de libros por el título del mismo.</div>		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 7. RUC-04. Buscar libro

RUC-05			
Titulo	Visualizar catálogo		
Descripción	<div>El usuario podrá visualizar el catálogo de libros disponibles.</div> <ul style="list-style-type: none">• Libro por categoría• Top de ventas• Libros que son novedad		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 8. RUC-05. Visualizar catálogo

RUC-06			
Titulo	Visualizar libro disponible.		
Descripción	<div>El usuario podrá acceder a la página de un libro disponible donde podrá:</div> <ul style="list-style-type: none">• Ver detalles del libro• Añadir libro al carro del pedido		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 9. RUC-06. Visualizar libro disponible

RUC-07			
Titulo	Ordenar libros		
Descripción	<div>El usuario podrá ordenar los libros disponibles en el catálogo:</div> <ul style="list-style-type: none">• Ordena alfabéticamente de forma ascendente.• Ordena alfabéticamente de forma descendente.• Ordenar de menor a mayor precio• Ordenar de mayor a menor precio.		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 10. RUC-07. Ordenar libros.

RUC-08			
Titulo	Pre-visualizar carrito		
Descripción	Pre-visualizar carrito de pedido para conocer el listado de libros añadidos.		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 11. RUC-08. Pre-visualizar carrito.

RUC-09			
Titulo	Mensaje de contacto		
Descripción	El usuario podrá enviar un mensaje de contacto a los administradores de la web.		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 12. RUC-09. Mensaje de contacto.

RUC-10			
Titulo	Salir de la web		
Descripción	Cerrar la sesión del usuario y salir de la web.		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 13. RUC-10. Salir de la web.

3.2.2 Requisitos de restricción

RUR-01			
Titulo	Idioma		
Descripción	El idioma de la aplicación será español		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 14. RUR-01. Idioma

RUR-02			
Titulo	Resolución interfaz de la web.		
Descripción	La aplicación web se adaptara a cualquier tipo de pantalla en la cual sea ejecutada.		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 15. RUR-02. Resolución interfaz web.

RUR-03			
Titulo	Navegador		
Descripción	Las paginas creadas serán compatibles con los navegadores web más comunes de ordenador(Mozilla Firefox, Google Chrome y Safari)		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 16. RUR-03. Navegador

RUR-04			
Titulo	Interfaz		
Descripción	La interfaz será clara y fácil de utilizar mediante el uso de iconos descriptivos, fuentes y controles adaptados y fácil navegación.		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 17. RUR-04. Interfaz.

3.3 Casos de uso

Los casos de uso especifican qué hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario. Con esta técnica podemos describir de una manera más clara lo que va a hacer el sistema.

En este apartado se mostrará una descripción tanto visual como textual de los casos de uso del sistema. A partir de los siguientes diagramas de casos de uso de UML se podrá comprender el funcionamiento del sistema además de conocer los distintos actores que pueden intervenir.

3.3.1 Diagramas UML

3.3.1.1 Diagramas administrador

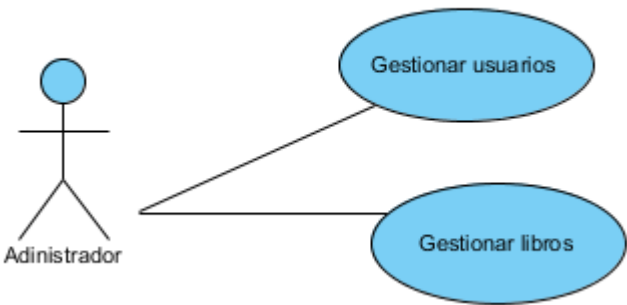


Ilustración 19. Casos de uso, usuario Administrador

Tal como se muestra en la Figura y veremos más detalladamente en el siguiente subapartado, el usuario administrador del sistema podrá gestionar tanto a los usuarios como a los libros de dichos usuarios.

3.3.1.2 Diagrama usuario



Ilustración 20. Casos de uso. Usuario

Cabe destacar que se considera que un usuario puede estar registrado o no en el sistema. El usuario registrado, como puede observarse en la imagen, puede acceder a las secciones privadas, que el usuario no registrado no puede acceder. Por tanto un usuario no registrado podría: registrarse, enviar un mensaje de contacto, buscar libros, ver catálogo de libros disponibles, ver detalles de cada libro y ordenar los libros del catálogo.

3.3.2 Descripción contextual

Para realizar de una forma ordenada la descripción textual se utilizará una plantilla. La plantilla es la siguiente:

Identificador	
Nombre	
Actor	
Descripción	
Flujo	
Precondiciones	
Postcondiciones	

Tabla 18. Plantilla casos de uso genérico.

A continuación se procede a describir detalladamente cada caso de uso, de modo que queden identificadas las interacciones entre el actor y el sistema, las condiciones que se precisan para que pueda darse dicho caso de uso, así como el flujo de interacción requerido para ser efectuado. Estas descripciones han sido desarrolladas de acuerdo a la plantilla mostrada en la Tabla 18, que se descompone en:

- Identificador: Código que identifica de manera única cada caso de uso. La nomenclatura se basa en el formato CUY-XX, donde Y es el actor (administrador o usuario) y XX es un número correlativo incremental, respecto de cada caso de uso.
- Nombre: Nombre descriptivo de cada caso de uso.
- Actor: Nombre del actor que interactúa con el sistema mediante el caso de uso especificado.
- Descripción: Conciso detalle sobre el funcionamiento del caso de uso.
- Flujo: Breve enumeración de los pasos intermedios que precisa el caso de uso para ser ejecutado.
- Precondiciones: Condiciones previas que son requeridas para que el caso de uso pueda llevarse a cabo.
- Postcondiciones: Condiciones que se darán una vez finalizada la ejecución del caso de uso.

3.3.2.1 Casos de uso: Usuario administrador.

CUA-01	
Nombre	Gestionar usuarios
Actor	Usuario administrador
Descripción	El administrador del sistema podrá eliminar usuarios.
Flujo	<div><div>1. Acceso a la aplicación destinada a la gestión de la página web.</div><div>2. Introducir datos de acceso</div><div>3. Navegar hasta la pestaña “Gestión de usuarios”.</div><div>4. Eliminar el usuario deseado.</div></div>
Precondiciones	El administrador ha de tener implantado el sistema.
Postcondiciones	CUA-02

Tabla 19. CUA-01. Gestionar usuarios.

CUA-02	
Nombre	Gestionar libros
Actor	Usuario administrador
Descripción	El administrador del sistema podrá eliminar los libros asociados a un usuario eliminado.
Flujo	<div><div>1. Acceso a la aplicación destinada a la gestión de la página web.</div><div>2. Introducir datos de acceso</div><div>3. Navegar hasta la pestaña “Gestión de libros”.</div><div>4. Eliminar los libros asociados al usuario eliminado.</div></div>
Precondiciones	CUA-01
Postcondiciones	Ninguna

Tabla 20. CUA-02. Gestionar libros.

3.3.2.2 Casos de uso: Usuario.

CUU-01	
Nombre	Registrarse
Actor	Usuario
Descripción	El usuario se registra en la web.
Flujo	<div><div>1. El usuario accede a la web</div><div>2. Accede al formulario de registro</div><div>3. Inserta todos los campos del formulario correctamente</div><div>4. Pulsa en “Registrarse”</div></div>
Precondiciones	Acceder a la web
Postcondiciones	Ninguna

Tabla 21. CUU-01. Registrarse.

CUU-02	
Nombre	Usuario Conectado
Actor	Usuario
Descripción	El usuario se conecta con sus credenciales a la web.
Flujo	<div><div>1. El usuario accede a la web</div><div>2. Introduce su nombre</div><div>3. Introduce su contraseña</div><div>4. Pulsa “Entrar” y se conectaría a la web.</div></div>
Precondiciones	CUU-01
Postcondiciones	CUU-08, CUU-09, CUU-10, CUU-11, CUU-12, CUU-13, CUU-14, CUU-15

Tabla 22. CUU-02. Usuario conectado.

CUU-03	
Nombre	Usuario no conectado
Actor	Usuario
Descripción	El usuario no está conectado en la web
Flujo	<div>1. El usuario accede a la web.</div> <div>2. Navega por la zona de usuario no registrados.</div>
Precondiciones	Ninguna
Postcondiciones	Ninguna

Tabla 23. CUU-03. Usuario no conectado.

CUU-04	
Nombre	Buscar libros
Actor	Usuario
Descripción	El usuario tiene la posibilidad de buscar un libro en la web
Flujo	<div>1. El usuario accede a la web.</div> <div>2. Introduce los valores de su búsqueda</div> <div>3. Pulsa el botón “Buscar”</div>
Precondiciones	Ninguna
Postcondiciones	Ninguna

Tabla 24. CUU-04. Buscar libros

CUU-05	
Nombre	Ver catálogo de libros disponibles
Actor	Usuario
Descripción	El usuario puede acceder al catálogo de libros disponibles
Flujo	<div>1. El usuario accede a la web.</div> <div>2. Accede a la zona de libros y le aparecen un catálogo con todos los libros disponibles en la web.</div>
Precondiciones	Ninguna
Postcondiciones	CUU-06

Tabla 25. CUU-05. Ver catálogo libros disponibles.

CUU-06	
Nombre	Ver detalles de libro
Actor	Usuario
Descripción	El usuario accede al catálogo de libros y selecciona uno de ellos.
Flujo	<ol style="list-style-type: none">1. El usuario accede a la web.2. Accede a la zona de libros y le aparecen un catálogo con todos los libros disponibles en la web3. Selecciona el libro deseado y pulsa en “Ver detalles”4. Le aparecen una serie de detalles asociados al libro elegido.
Precondiciones	CUU-05
Postcondiciones	Ninguna

Tabla 26. CUU-06. Ver detalles libro.

CUU-07	
Nombre	Ordenar libros
Actor	Usuario
Descripción	El usuario accede al catálogo de libros y los ordena por el orden seleccionado.
Flujo	<ol style="list-style-type: none">1. El usuario accede a la web.2. Accede a la zona de libros y le aparecen un catálogo con todos los libros disponibles en la web3. Elige y pulsa en el botón “Ordenar por” el orden deseado.
Precondiciones	CUU-05
Postcondiciones	Ninguna

Tabla 27. CUU-07. Ordenar libros.

CUU-08	
Nombre	Acceso menú personal
Actor	Usuario
Descripción	El usuario accede a su menú personal
Flujo	<div><div>1. El usuario accede a la web</div><div>2. Introduce su nombre</div><div>3. Introduce su contraseña</div><div>4. Pulsa “Entrar” y se conectaría a la web.</div><div>5. Una vez dentro accede a la sección “Mi cuenta”</div></div>
Precondiciones	CUU-02
Postcondiciones	CUU-09, CUU-10, CUU-11, CUU-12, CUU-13, CUU-14

Tabla 28. CUU-08. Acceso menú personal.

CUU-09	
Nombre	Ver datos personales
Actor	Usuario
Descripción	El usuario accede a sus datos personales
Flujo	<div><div>1. El usuario accede a la web</div><div>2. Introduce su nombre</div><div>3. Introduce su contraseña</div><div>4. Pulsa “Entrar” y se conectaría a la web.</div><div>5. Una vez dentro accede a la sección “Mi cuenta”.</div><div>6. Accede a la subsección “Mis datos personales”</div></div>
Precondiciones	CUU-08
Postcondiciones	Ninguna

Tabla 29. CUU-09. Ver datos personales.

CUU-10	
Nombre	Modificar datos personales
Actor	Usuario
Descripción	El usuario accede al menú personal para modificar sus datos personales.
Flujo	<div><div>1. El usuario accede a la web</div><div>2. Introduce su nombre</div><div>3. Introduce su contraseña</div><div>4. Pulsa “Entrar” y se conectaría a la web.</div><div>5. Una vez dentro accede a la sección “Mi cuenta”.</div><div>6. Accede a la subsección “Editar perfil”</div><div>7. Introduce los valores que quiere cambiar en el formulario de actualización de datos personales.</div><div>8. Pulsa botón “Actualizar”</div></div>
Precondiciones	CUU-08
Postcondiciones	Ninguna

Tabla 30. CUU-10. Modificar datos personales.

CUU-11	
Nombre	Ver listado de pedidos
Actor	Usuario
Descripción	El usuario accede a su historial de pedido en la web
Flujo	<div><div>1. El usuario accede a la web</div><div>2. Introduce su nombre</div><div>3. Introduce su contraseña</div><div>4. Pulsa “Entrar” y se conectaría a la web.</div><div>5. Una vez dentro accede a la sección “Mi cuenta”.</div><div>6. Accede a la subsección “Mis pedidos”</div></div>
Precondiciones	CUU-08
Postcondiciones	Ninguna

Tabla 31. CUU-11. Ver listado pedidos.

CUU-12	
Nombre	Ver carro pedido actual
Actor	Usuario
Descripción	El usuario accede a su carro de pedido actual
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web 2. Introduce su nombre 3. Introduce su contraseña 4. Pulsa "Entrar" y se conectaría a la web. 5. Una vez dentro accede a la sección "Mi cuenta". 6. Accede a la subsección "Mi carrito".
Precondiciones	CUU-08
Postcondiciones	Ninguna

Tabla 32. CUU-12. Ver carro pedido actual.

CUU-13	
Nombre	Subir libro
Actor	Usuario
Descripción	El usuario accede al formulario de subida de libros.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la web 2. Introduce su nombre 3. Introduce su contraseña 4. Pulsa "Entrar" y se conectaría a la web. 5. Una vez dentro accede a la sección "Mi cuenta". 6. Accede a la subsección "Subir libro". 7. Rellena el formulario de subida de libros. 8. Pulsa en "Subir libro".
Precondiciones	CUU-08
Postcondiciones	Ninguna

Tabla 33. CUU-13. Subir libro.

CUU-14	
Nombre	Mis libros
Actor	Usuario
Descripción	El usuario accede a historial de libros subidos a la web.
Flujo	<div><div>1. El usuario accede a la web</div><div>2. Introduce su nombre</div><div>3. Introduce su contraseña</div><div>4. Pulsa “Entrar” y se conectaría a la web.</div><div>5. Una vez dentro accede a la sección “Mi cuenta”.</div><div>6. Accede a la subsección “Mis libros”.</div></div>
Precondiciones	CUU-08
Postcondiciones	Ninguna

Tabla 34. CUU-14. Mis libros.

CUU-15	
Nombre	Añadir libro al carro
Actor	Usuario
Descripción	El usuario mientras realiza un pedido añade libros a su carro.
Flujo	<div><div>1. El usuario accede a la web</div><div>2. Introduce su nombre</div><div>3. Introduce su contraseña</div><div>4. Pulsa “Entrar” y se conectaría a la web.</div><div>5. Accede al catálogo de libros disponibles</div><div>6. Selecciona el libro que quiere</div><div>7. Pulsa “Añadir al carro”</div></div>
Precondiciones	CUU-02
Postcondiciones	Ninguna

Tabla 35. CUU-15. Añadir libro al carro.

CUU-16	
Nombre	Eliminar libro del carro
Actor	Usuario
Descripción	El usuario mientras realiza un pedido elimina libros de su carro.
Flujo	<ol style="list-style-type: none">1. El usuario accede a la web2. Introduce su nombre3. Introduce su contraseña4. Pulsa “Entrar” y se conectaría a la web.5. Accede al catálogo de libros disponibles6. Selecciona el libro que quiere7. Pulsa “Añadir al carro”.8. Accede a su menú personal y en la sección “Mi carrito actual”.9. En el listado de artículos existentes, se puede eliminar los libros que se deseen.
Precondiciones	CUU-02
Postcondiciones	Ninguna

Tabla 36. CUU-16. Eliminar libro del carro.

CUU-17	
Nombre	Enviar mensaje de consulta
Actor	Usuario
Descripción	El usuario puede acceder a la web y enviar un mensaje de consulta.
Flujo	<ol style="list-style-type: none">1. El usuario accede a la web2. En la sección “Contacto”3. Rellena el formulario de consulta4. Pulsa “Enviar”
Precondiciones	Ninguna
Postcondiciones	Ninguna

Tabla 37. CUU-17. Enviar mensaje de consulta.

CUU-18	
Nombre	Cerrar sesión
Actor	Usuario
Descripción	El usuario puede abandonar su sesión en la web.
Flujo	<div>1. El usuario conectado</div> <div>2. Pulsa “Salir” y cerraría su sesión en la web.</div>
Precondiciones	CUU-02
Postcondiciones	Ninguna

Tabla 38. CUU-18. Cerrar sesión.

3.4 Requisitos de software

Una vez obtenidos los requisitos de usuario y los casos de uso se puede obtener la lista de los requisitos software, que serán la base fundamental de lo que va a ser finalmente la aplicación.

Estos requisitos se dividen en dos grupos fundamentales: los requisitos funcionales y los requisitos no funcionales. Se utilizará la misma plantilla que para los requisitos de usuario, lo único que les diferenciará será su identificador y el campo “fuente” que se sustituirá por dependencia. En este caso los requisitos funcionales será RSF y los no funcionales serán RSNF. El campo dependencia designará los requisitos de capacidad con los que comparte algún tipo de dependencia.

Identificador			
Título			
Descripción			
Dependencia		Necesidad	
Prioridad		Claridad	
Estabilidad		Verificabilidad	

Tabla 39. Plantilla RSF y RSNF.

3.4.1 Requisitos funcionales

Son aquellos requisitos que describen el funcionamiento del sistema. De este modo, definirán el comportamiento interno del software especificando cómo los casos de uso serán llevados a cabo.

RSF-01			
Titulo	Registro de usuario		
Descripción	El sistema deberá mostrar un formulario de registro para que el usuario pueda registrarse.		
Dependencia	RUC-01	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 40. RSF-01. Registro de usuario.

RSF-02			
Titulo	Acceso a la web		
Descripción	El sistema mostrará un formulario de acceso, para que el usuario pueda introducir sus credenciales y acceder a la web.		
Dependencia	RUC-02	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 41. RSF-02. Acceso a la web.

RSF-03			
Titulo	Acceso a los datos personales		
Descripción	El usuario deberá poder acceder a sus datos personales, que fueron añadidos cuando se registró en la web.		
Dependencia	RUC-03	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 42. RSF-03. Accesos a los datos personales.

RSF-04			
Titulo	Modificar datos personales		
Descripción	El usuario deberá poder modificar y actualizar sus datos personales en la base de datos de la aplicación.		
Dependencia	RUC-03	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 43. RSSF-04. Modificar datos personales.

RSF-05			
Titulo	Subir libros		
Descripción	El usuario deberá poder subir libros al catálogo de la web, utilizando el formulario que le muestra el sistema para la subida de libro.		
Dependencia	RUC-03	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 44. RSF-05. Subir libros

RSF-06			
Titulo	Historial de libros subidos		
Descripción	El usuario deberá poder acceder a ver su historial de libros subidos a la web.		
Dependencia	RUC-03	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 45. RSF-06. Historial de libros subidos.

RSF-07			
Titulo	Historial de pedidos		
Descripción	El usuario deberá poder acceder a su historial de pedidos que haya realizado en la web.		
Dependencia	RUC-03	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 46. RSF-07. Historial de pedidos.

RSF-08			
Titulo	Carrito de pedido actual.		
Descripción	El usuario deberá poder acceder a la información acerca de su carrito de pedido actual.		
Dependencia	RUC-03	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 47. RSF-08. Carrito de pedido actual.

RSF-09			
Titulo	Buscar libro		
Descripción	El usuario dispondrá de un buscador de títulos de libros.		
Dependencia	RUC-04	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 48. RSF-09. Buscar libro.

RSF-10			
Titulo	Libros por categorías		
Descripción	El usuario podrá listar el catalogo por cada una de las categorías existentes.		
Dependencia	RUC-05	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 49. RSF-10. Libros por categorías.

RSF-11			
Titulo	Top ventas		
Descripción	El usuario podrá ver que libros son los más vendidos en la web.		
Dependencia	RUC-05	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 50. RSF-11. Top ventas.

RSF-12			
Titulo	Novedades		
Descripción	El usuario podrá tener acceso a las novedades añadidas al catálogo de libros disponibles.		
Dependencia	RUC-05	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 51. RSF-12. Novedades.

RSF-13			
Titulo	Ver detalles del libro		
Descripción	El usuario podrá acceder a la información individual de cada libro.		
Dependencia	RUC-06	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 52. RSF-13. Ver detalles del libro.

RSF-14			
Titulo	Añadir libro al carro del pedido		
Descripción	El usuario podrá añadir libros al carrito del pedido		
Dependencia	RUC-06	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 53. RSF-14. Añadir libro al carro del pedido.

RSF-15			
Titulo	Ordenar libros alfabéticamente		
Descripción	El usuario podrá ordenar alfabéticamente los libros del catálogo de forma ascendente.		
Dependencia	RUC-07	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 54. RSF-15. Ordenar libros alfabéticamente I.

RSF-16			
Titulo	Ordenar libros alfabéticamente.		
Descripción	El usuario podrá ordenar alfabéticamente los libros del catálogo de forma descendente.		
Dependencia	RUC-07	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 55. RSF-16. Ordenar libros alfabéticamente II.

RSF-17			
Titulo	Ordenar libros por precio		
Descripción	El usuario podrá ordenar los libros del catálogo por precio de forma ascendente.		
Dependencia	RUC-07	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 56. RSF-17. Ordenar libros por precio I.

RSF-18			
Titulo	Ordenar libros por precio		
Descripción	El usuario podrá ordenar los libros del catálogo por precio de forma descendente.		
Dependencia	RUC-07	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 57. RSF-18. Ordenar libros por precio II.

RSF-19			
Titulo	Eliminar libro del carrito		
Descripción	El usuario podrá eliminar libros de su carrito actual.		
Dependencia	RUC-08	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 58. RSF-19. Eliminar libro del carrito.

RSF-20			
Titulo	Envío de mensaje de contacto		
Descripción	El usuario podrá enviar un mensaje de contacto a través del formulario que le muestra el sistema.		
Dependencia	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 59. RSF-20. Envío de mensaje de contacto.

RSF-21			
Titulo	Salir de la web		
Descripción	El usuario podrá cerrar su sesión y salir de la web.		
Dependencia	RUC-10	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 60. RSF-21. Salir de la web.

3.4.2. Requisitos no funcionales

Debido a que el cliente y el desarrollador son la misma persona en este caso, los requisitos no funcionales no darán más información que los requisitos de restricción y por lo tanto son los mismos. Ahora se volverán a adjuntar organizados por categorías.

Acrónimo	Tipo Requisito
RSNFI	Requisito no Funcional de Interfaz
RSNFO	Requisito No funcional de Operación

Tabla 61. Acrónimos Requisitos No funcionales.

3.4.2.1. Requisitos de interfaz

RSNFI-01			
Titulo	Idioma		
Descripción	El idioma de la aplicación será español		
Dependencia	RUR-01	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 62. RSNFI-01. Idioma.

RSNFI-02			
Titulo	Resolución interfaz de la web.		
Descripción	La interfaz de la aplicación web se adaptara a cualquier tipo de pantalla ya sea cual sea el tamaño, donde sea ejecutada.		
Dependencia	RUR-02	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 63. RSNFI-02. Resolución interfaz de la web.

RSNFI-03			
Titulo	Interfaz		
Descripción	La interfaz será clara y fácil de utilizar mediante el uso de iconos descriptivos, fuentes y controles adaptados y fácil navegación.		
Dependencia	RUR-04	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 64. RSNFI-03. Interfaz.

3.4.2.2. *Requisitos de operación*

RSNFO-03			
Titulo	La web funcionara en diferente tipo de navegadores		
Descripción	La web se ejecutará en todos los navegadores web utilizados en la actualidad		
Fuente	Cliente.	Necesidad	Alta.
Prioridad	Alta.	Claridad	Alta.
Estabilidad	Estable.	Verificabilidad	Alta.

Tabla 65. RSNFO-04. Navegadores.

4. CAPÍTULO 4 DISEÑO

4.1. Introducción

En este capítulo se van a tratar las fases de diseño e implementación de la solución elegida. Se van a presentar el diseño de arquitectura como las herramientas que se han utilizado y se mostrará un prototipo final del resultado de la propuesta de solución.

4.2. Diseño: Arquitectura

En este apartado se presenta el modelo elegido para explicar la relación entre los distintos componentes que lo forman.

Para este proyecto se ha decidido utilizar la arquitectura modelo-vista-controlador. Que permite de forma intuitiva distribuir por capas el proyecto, facilitando el acceso a componentes concretos que requieran de una modificación o arreglo, garantizando el correcto funcionamiento del resto.

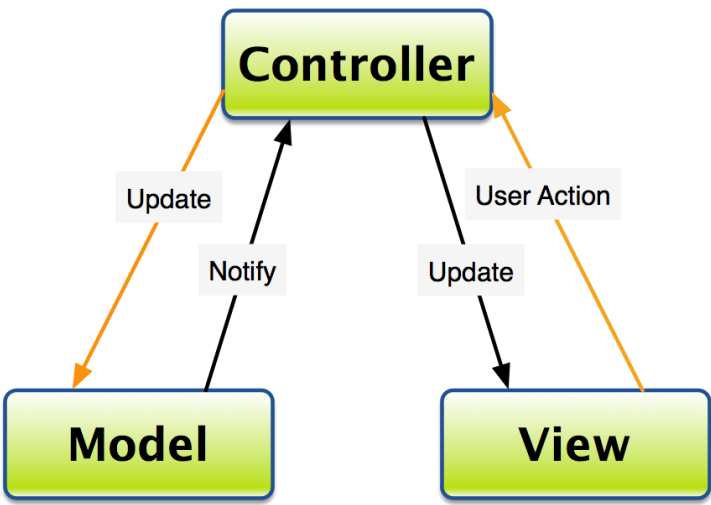


Ilustración 21. Arquitectura Model-View-Controller

Como se puede ver en la Figura anterior, nuestro modelo está formado por tres capas que se corresponden con las indicadas en la imagen.

En la imagen siguiente se puede ver el diagrama de clases del proyecto. Se puede diferenciar cada uno de las capas de la arquitectura y que corresponden a cada uno de las partes del sistema:

La primera, la vista, se corresponde con la interfaz que da el sistema, es decir, la vista que percibe el usuario cuando accede a la web y cómo va a poder interactuar con ella a la hora de resolverlo.

La segunda capa, el controlador, representa la lógica del sistema que está en la parte del servidor y realizará las funciones que sean necesarias para llevar a cabo el objetivo del sistema como puede ser gestionar usuarios o control de sesiones entre usuarios. En esta parte del sistema es donde está la mayor parte del trabajo ya que hay que realizar toda la funcionalidad.

Por último, el modelo es la parte en la que se trabaja con la base de datos, por tanto en nuestro caso se mostrara la especificación de cada una de las bases de datos utilizadas, la de MySQL y la de MongoDB.



Ilustración 22. Esquema de la arquitectura diseñada.

4.2.1. View

Esta es la capa que se encarga de mostrar al usuario una interfaz gráfica, ofreciéndole la posibilidad de interactuar con ella. Presenta la información la función del rol del usuario utilizando distintos componentes visuales, proporcionándole un medio de enlace con el controlador.

Cuando el usuario interactúa con la interfaz, la información que introduce, ya sean eventos por pulsaciones de botones o datos introducidos, se comunican al controlador a través de Socket IO [29], para que éste pueda notificar al modelo y se realicen los cambios pertinentes. Para esta comunicación, el componente de la vista utiliza las vistas proporcionadas por el controlador mediante un intérprete HTML que se especificará posteriormente.

4.2.2. Controller

Es la capa encargada de responder a eventos (acciones del usuario) e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información, pudiendo enviar comandos a la vista asociada si se solicitara un cambio en la forma en la que se presenta el modelo.

Por tanto se podría decir que esta capa hace de intermediaria entre la capa “view” y la capa “model”.

4.2.2.1. *Controlador - Core*

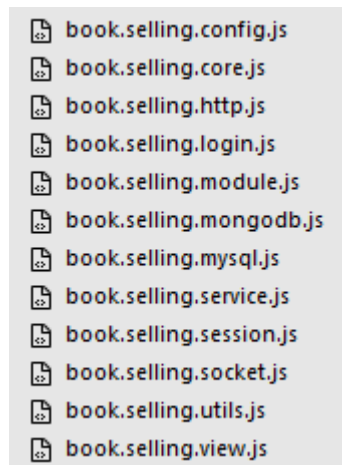


Ilustración 23. Controlador de BookSelling

En la Ilustración 19, se pueden observar todas las clases encargadas de controlar todas las peticiones que se realizan en la vista. Ahora se procederá a comentar la función empeñada por cada una de ellas:

- **Book.selling.config.js**

Almacena todas las variables de configuración, que permiten la conectividad con las base de datos, así como ubicaciones en el sistema de ficheros del servidor.

- **Book.selling.core.js**

Genera la estructura inicial de la aplicación, dónde se van instanciando todas las extensiones, plugins y los complementos que permiten el correcto funcionamiento del servicio.

- **Book.selling.http.js**

Se encarga de las solicitudes Http, que los usuarios realizan desde sus dispositivos, controlando que peticiones están realizando y a qué servicio o vista quieren acceder.

- **Book.selling.login.js**

Gestiona el acceso y el registro de usuarios. Dos funciones destacables en este componente:

- “SignUp”: función utilizada para darse de alta como un usuario registrado.
- “SignIn”: función utilizada para identificarse como usuario registrado.

- **Book.selling.service.js**

Realiza la carga de los servicios existentes, “admin” o “site” dependiendo del rol del usuario, ofreciéndole la capacidad de modificar contenido de la página en función de sus permisos. Destacar la función “getViews”, que permite la carga de las vistas asociadas a ese servicio.

- **Book.selling.view.js**

Gestiona la carga de las diferentes vistas, compuestas por una serie de módulos. Distribuyendo éstos con un orden y una ubicación específica.

- **Book.selling.module.js**

Encargada de la carga de los distintos módulos. Éstos contienen un fragmento de vista y otro de controlador que permite generar dicha vista mediante los datos del modelo. Encapsulando la funcionalidad de dicho módulo, haciéndolo reutilizable para otras vistas.

- Module.render: función que interpreta mediante el controlador la vista.

- Module.Load: función que carga los diferentes módulos.
- Module.getFile: función que recupera el directorio del módulo.
- Module.getHTML: función que devuelve la vista HTML correspondiente.
- Module.getScript: función que devuelve el controlador correspondiente.

- **Book.selling.mongodb.js**

Administra las conexiones de MongoDB, permitiendo realizar las siguientes operaciones: update, insert, remove, save y replace.

- **Book.selling.mysql.js**

Administra las conexiones de mysql, así como las consultas que se pueden hacer sobre la base de datos.

- **Book.selling.session.js**

Clase destinada a gestionar la sesión de los usuarios, ofreciendo la posibilidad de que el usuario identificado permanezca por un periodo de tiempo determinado, conectado al servicio. Tiene las siguientes operaciones: get, set o delete.

- **Book.selling.socket.js**

Inicializa el socket, estableciendo una conexión bidireccional entre el cliente y el servidor, comunicando a través de eventos la capa “view” y la capa “controller”.

- **Book.selling.utils.js**

Almacena un conjunto de utilidades dispuestas globalmente para su uso en el controlador.

4.2.2.2. Twig

Interprete HTML que permite generar iteraciones, condicionales y mostrar salida de variables, para generar la vista [31]. Gracias a su conexión con el controlador.

4.2.2.3. Services/ Views / Modules

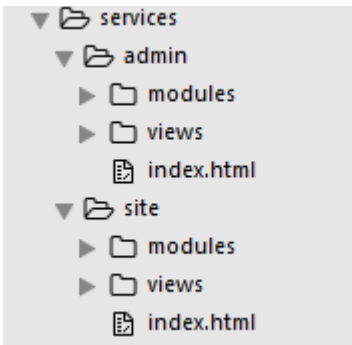


Ilustración 24. Servicios, vistas y módulos.

En la Ilustración 22, se pueden apreciar los servicios existentes, uno para la administración y otro para el sitio accesible por el usuario.

Como se puede observar la Ilustración 20, cada uno de los servicios posee en su interior una serie de vistas formadas por módulos.

4.2.3.Model

En la capa de modelo de la aplicación se ubican los dos componentes de almacenamiento de datos. Son los encargados de almacenar los datos e información del sistema garantizando su persistencia y disponibilidad. Estos componentes permite la comunicación con los subcomponentes del controlador proporcionándoles la posibilidad de almacenar y obtener información según sea necesario.

Como puede verse en la Ilustración 23, en esta capa existen dos unidades de persistencia. La primera de ellas SQL y otra NoSql. A continuación se mostrará un diagrama de cada una de ellas:

4.2.3.1. Diagrama entidad- relación bases de datos SQL.

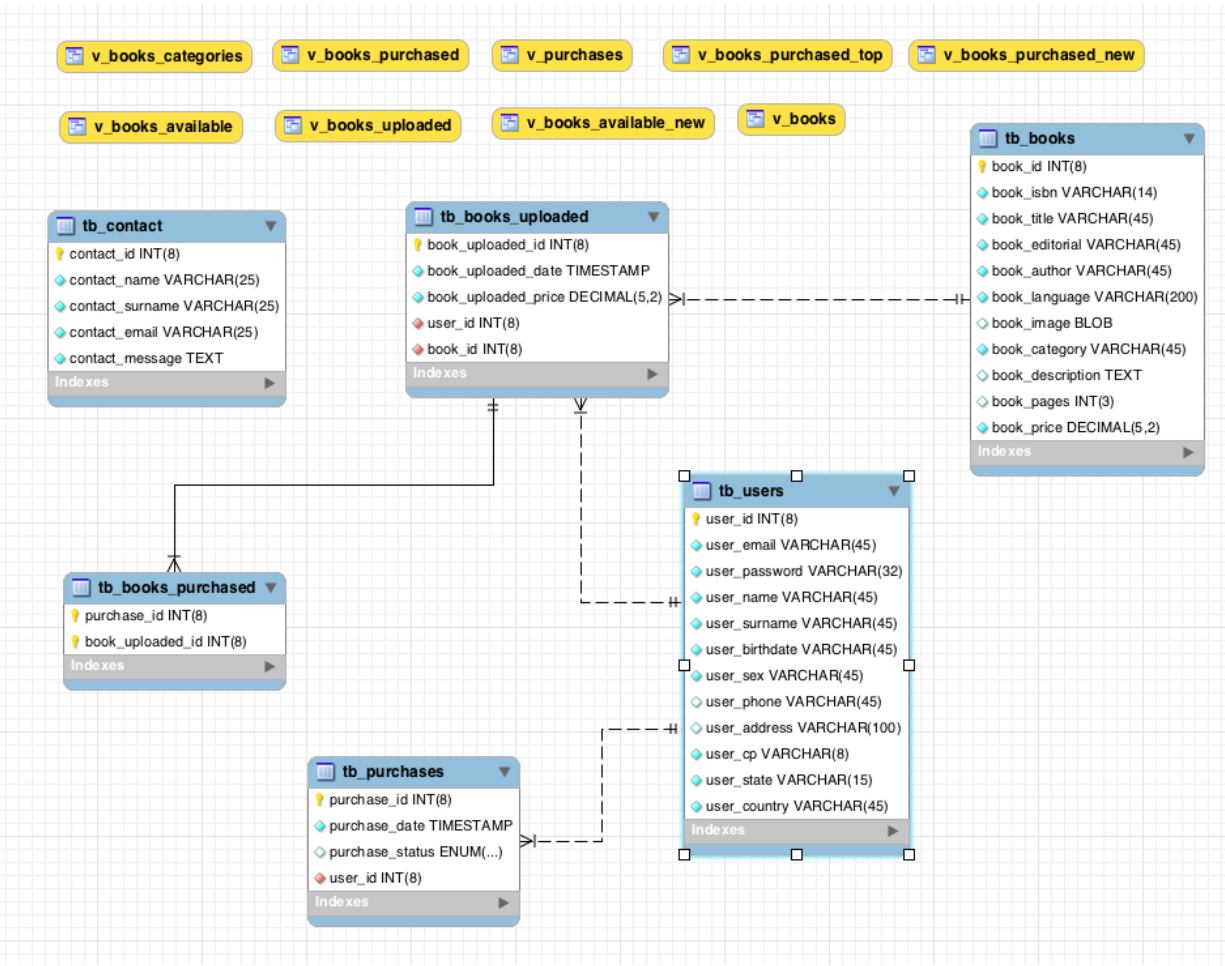


Ilustración 25: Modelo entidad-relacion. BookSelling

Se va a proceder a la explicación de cada una de las tablas de la Ilustración 21:

- **tb_books**: en esta tabla se almacena todo el catálogo de libros existente en la aplicación.
- **tb_users**: tabla encargada del almacenamiento de los datos de los usuarios registrados en la web.
- **tb_purchases**: tabla donde se almacenan los pedidos realizados y completados en la aplicación.
- **tb_books_uploaded**: en esta otra tabla se almacenan los libros subidos por los usuarios a la aplicación.
- **tb_books_purchased**: esta tabla muestra los libros subidos por los usuarios que han sido procesados en algún pedido.

- tb_contact: en esta otra tabla se almacenan los mensajes de consultas o duda enviados pos los visitantes de la aplicación.

Además de las tablas usadas en el diagrama entidad-relación, también han sido necesario la creación de vistas para facilitar el acceso a los datos y a la compresión de las consultas. A continuación se describirán las vistas utilizadas según han sido creadas en MySql:

- v_books_categories: en esta vista se muestran las diferentes categorías de libros en venta.
- v_books_purchased: muestra una lista de libros vendidos en los pedidos.
- v_purchases: muestra todos los pedidos realizados.
- v_books_purchased_top: muestra un listado el catálogo de libros más pedidos.
- v_books_purchased_new: muestra un listado con los últimos libros vendidos.
- v_books_available: muestra una lista con todos los libros disponibles en el catálogo.
- v_books_uploaded: muestra todos los libros subidos por los usuarios.
- v_books_available_new: muestra los últimos libros añadidos por los usuarios
- v_books: muestra todos los libros existentes en la bases de datos.

4.2.3.2. Diagrama NoSQL.

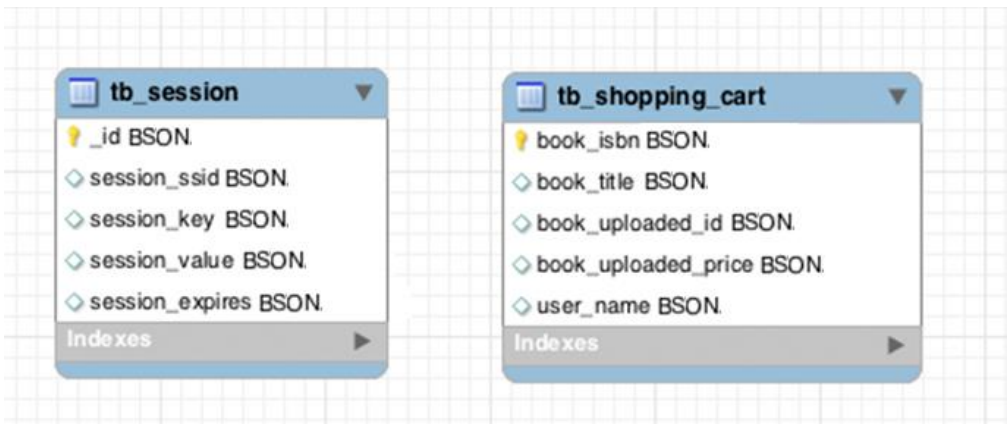


Ilustración 26: Sesión y carrito en MongoDB.

En la ilustración 24 se pueden observar las tablas existentes en la base de datos NoSql. Como puede verse todos los atributos almacenados son objetos BSON.

Se pueden distinguir dos tablas, la primera de ellas contiene todos los datos referentes a la sesión de usuario. La otra contiene la información relacionada con el carrito de pedido de libros que cada usuario almacena en cada sesión.

4.3. Implementación: Interfaz web final

4.3.1. Interfaz usuario

4.3.1.1. Inicio

El acceso a la web tiene como página de bienvenida la siguiente figura de más abajo, donde aparecerá un buscador en la zona superior derecha, justo debajo de él, una zona de acceso o registro. A la izquierda, también en la parte superior, se ve el logotipo de “BookSelling” junto a su eslogan. Debajo de todo esto se encuentra la barra de navegación, con las secciones existentes, variables en función de si existe usuario conectado a la web.

En el cuerpo de la página se puede ver una pasarela de libros y justo debajo de la pasarela, aparece el catálogo de libros disponibles.

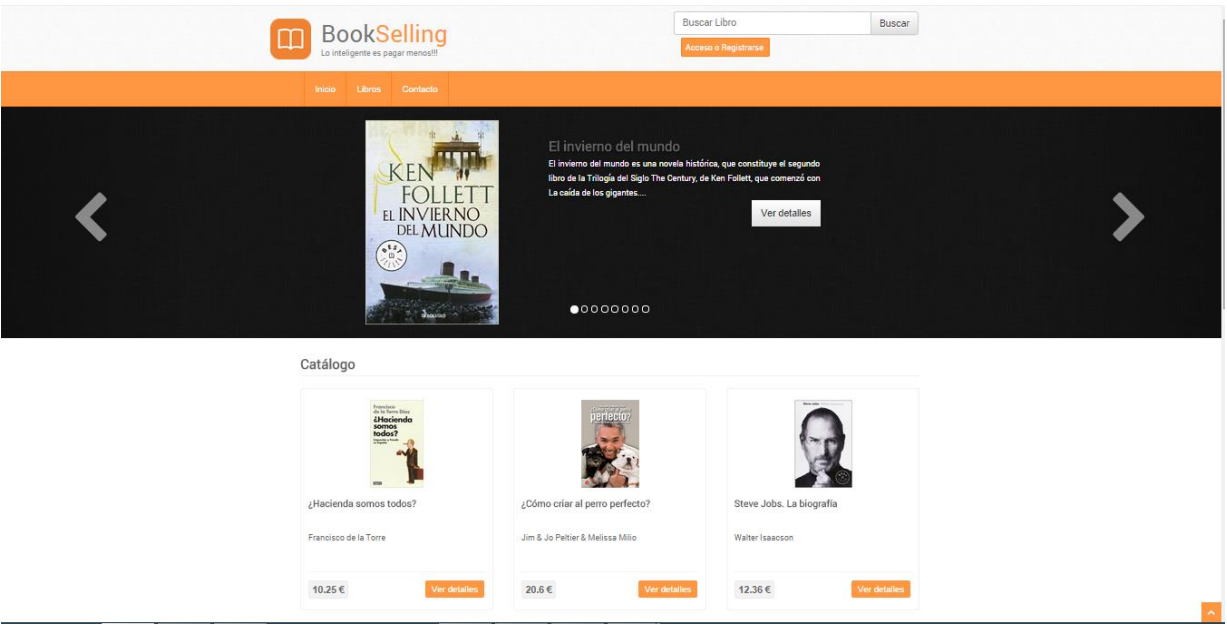


Ilustración 27: Página de inicio de la aplicación.

4.3.1.2. Menú personal

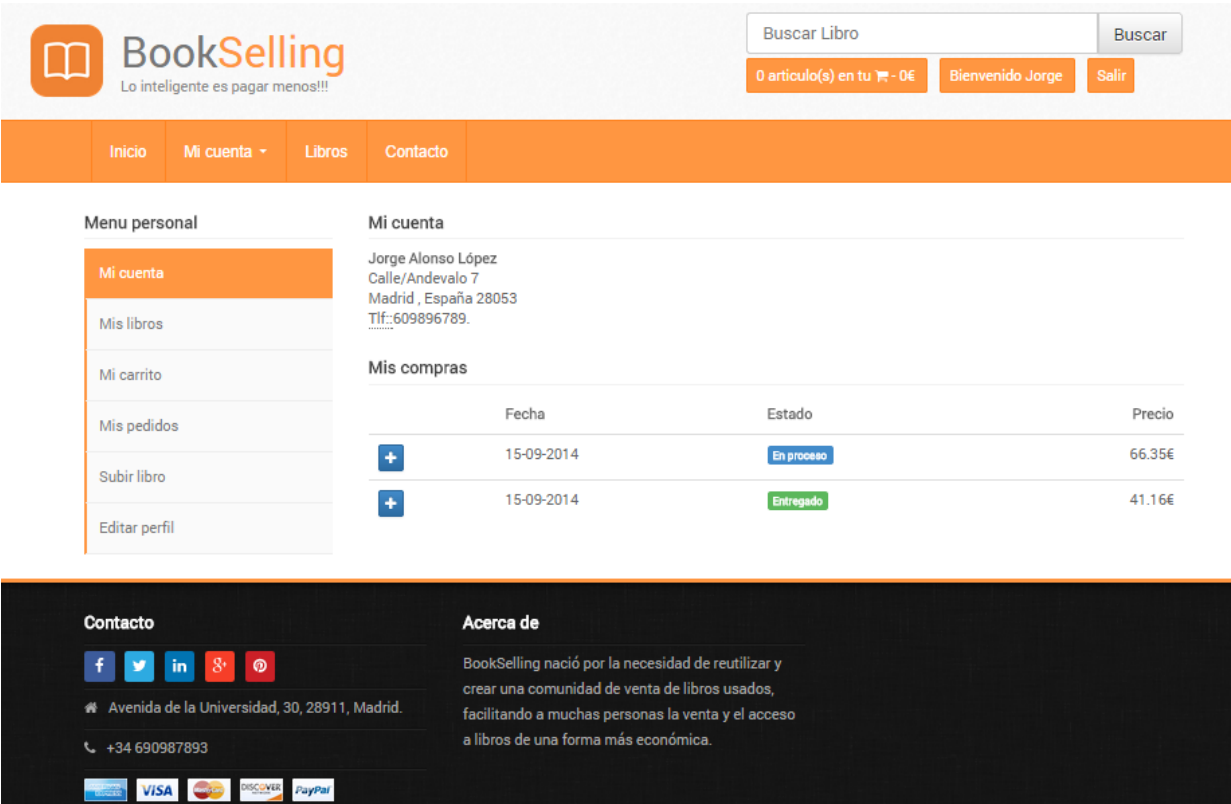


Ilustración 28. Menú personal del usuario.

En la Ilustración 26, se puede observar el menú personal de un usuario que ha accedido a la web. En él se pueden ver las diferentes opciones que ofrece el menú personal a los usuarios, permitiéndoles acceder a:

- Datos personales
- Datos del carrito actual
- Historial pedidos realizados
- Subir libro
- Editar los datos personales.

4.3.1.3. Buscador

En la siguiente imagen (Ilustración 27), puede verse el buscador implementado. El cual busca la cadena de texto introducida entre todos los títulos de libros existentes en el catálogo, sugiriendo todo aquel que la contenga, como puede observarse más abajo.

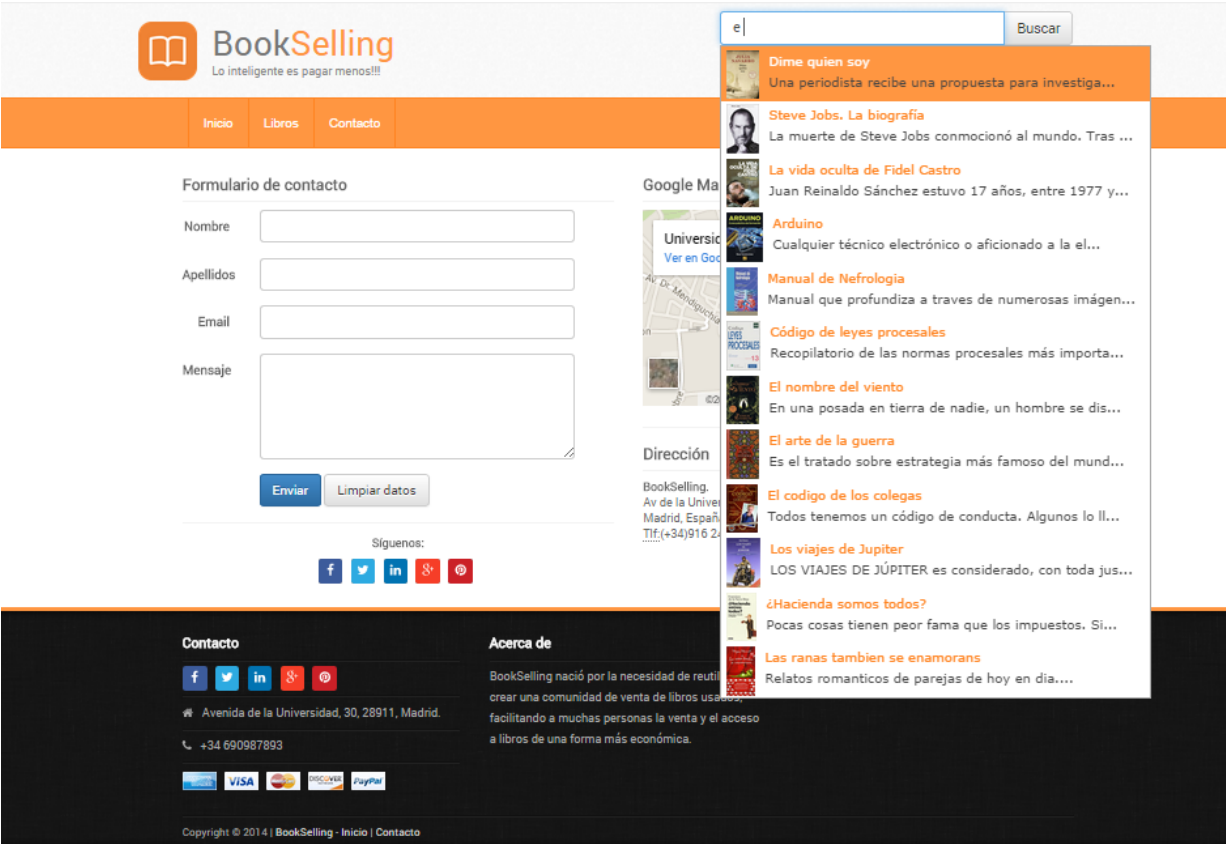


Ilustración 29. Buscador de libros de aplicación.

4.3.1.4. Acceso a la web

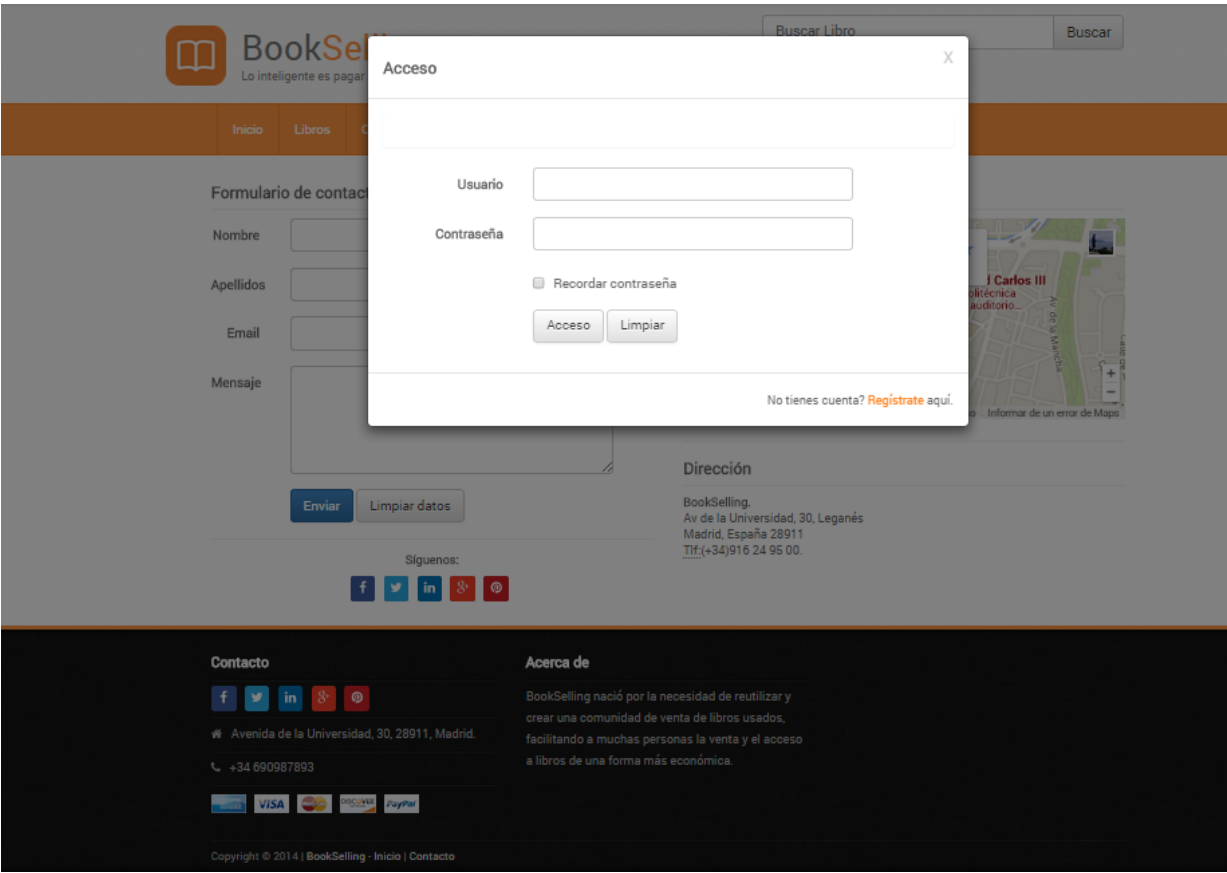


Ilustración 30. Página de acceso a la web

En la imagen superior, puede verse el formulario de acceso a la web. El usuario registrado introduce correctamente el “Usuario de acceso” y la “Contraseña” accedería a la web y al menú de los usuarios dados de alta en la web y a la opción de comprar libros.

4.3.1.5. Registro en la web

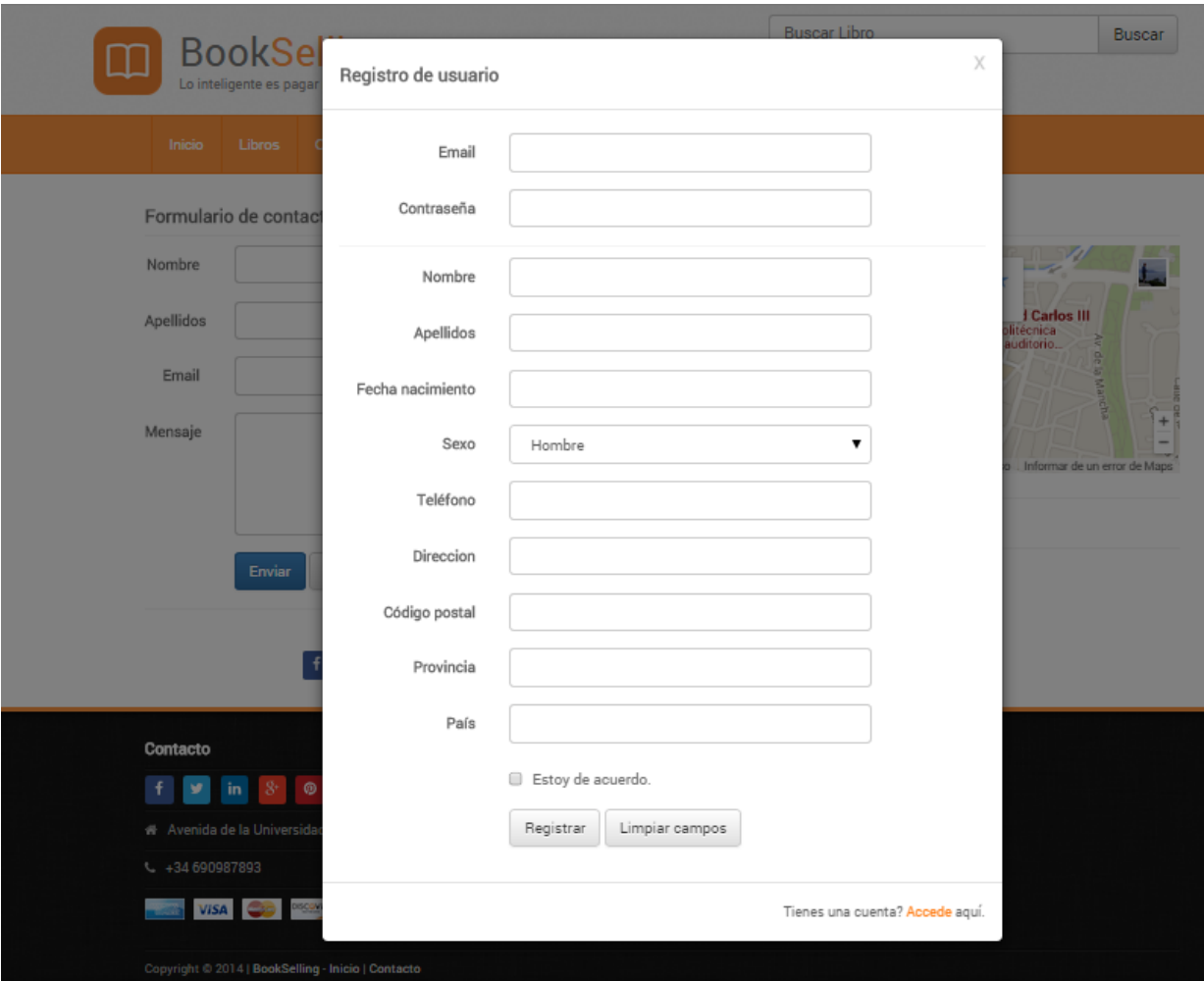


Ilustración 31. Página de registro en la web.

Todo usuario que acceda a la web y desee registrarse para disfrutar de todas las ventajas de ser usuario registrado, podrá hacerlo a través del formulario que se puede ver en la Ilustración 29. Basta con rellenarlo correctamente y el usuario en cuestión formará parte de los usuarios registrados de la web.

4.3.1.6. Libros

En la siguiente interfaz se puede observar, como se dispondrá el catálogo de os libros. Se podrán listar por:

- Todos los libros disponibles del catálogo.
- Acceder a los libros por una categoría específica.
- Acceder al Top 10 de libros vendidos
- Acceder a los libros buscados.

Además cada listado de libros mostrado se podrá ordenar:

- Alfabéticamente de la a-z
- Alfabéticamente de la z-a
- Por precio ascendente
- Por precio descendente

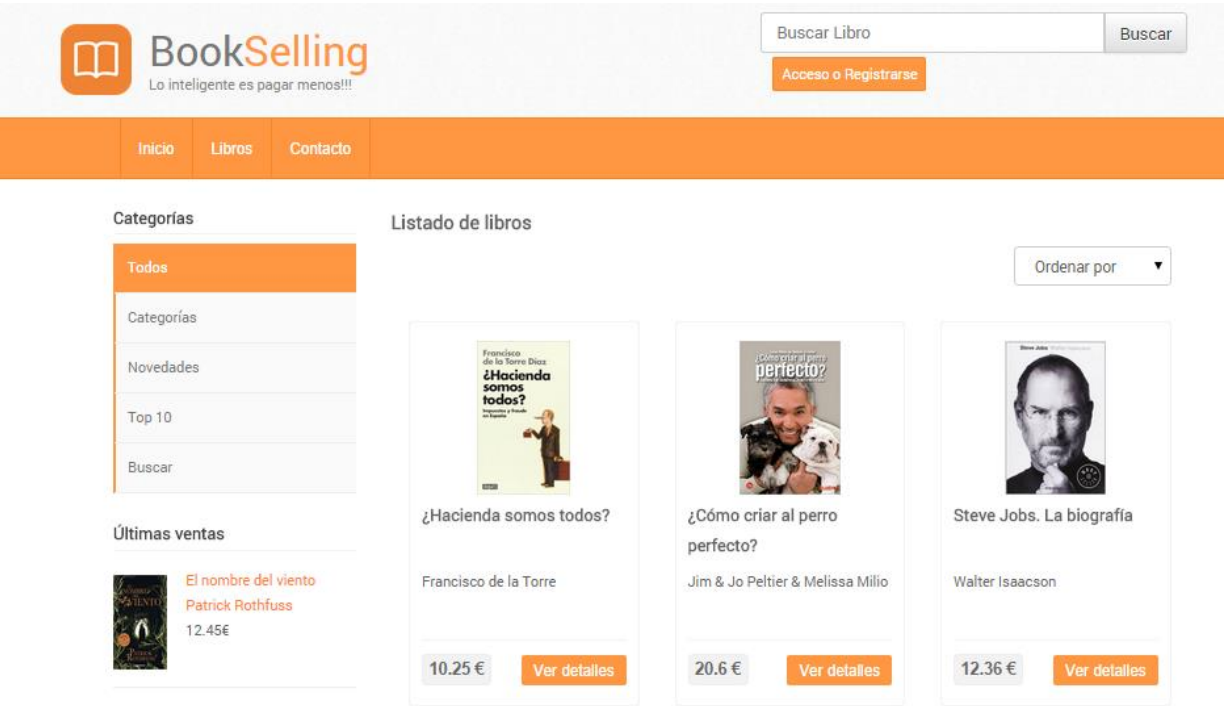


Ilustración 32. Libros mostrados por la aplicación.

4.3.1.7. Contacto

La web también tiene una sección de contacto, donde los usuarios podrán enviar a través del formulario que aparece en ella, toda duda o consulta que les surja. Más abajo se muestra la interfaz definitiva de la sección “Contacto”:

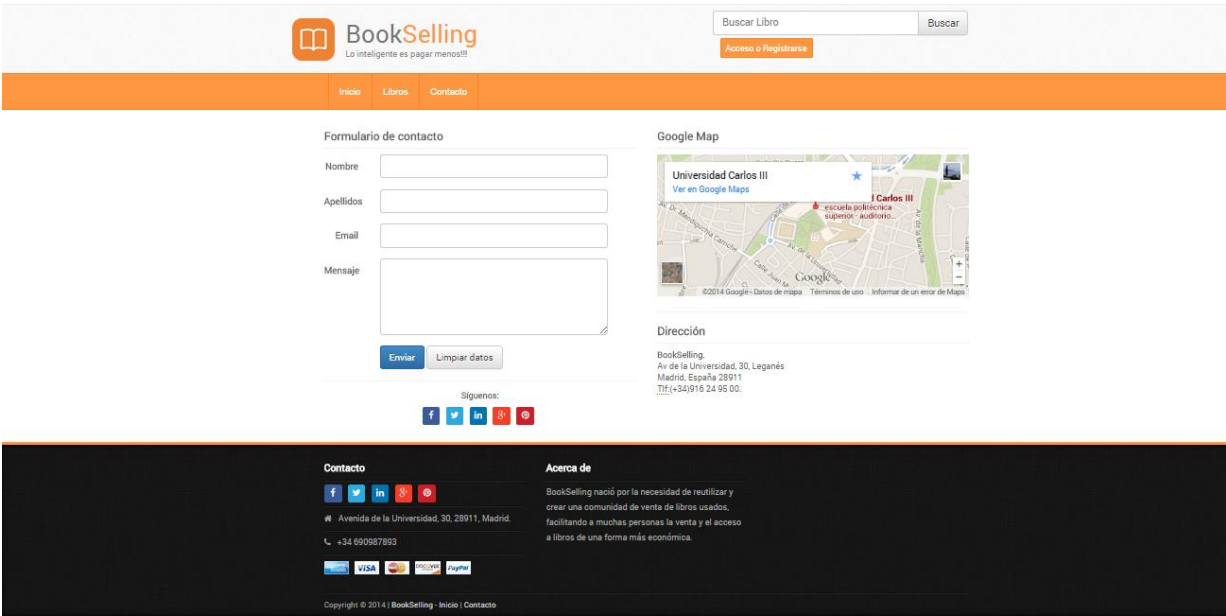


Ilustración 33: Página de contacto de la aplicación

4.3.2. Interfaz usuario administrador

4.3.2.1. Interfaz Usuario administrador.

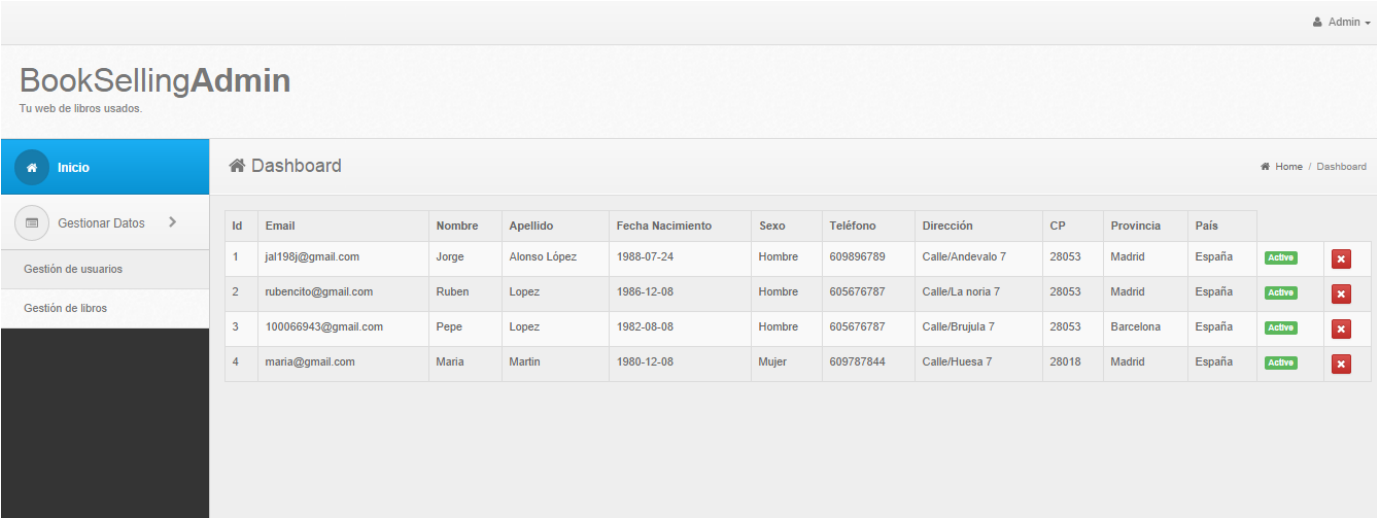


Ilustración 34. Administración. Gestión de usuarios.

En la imagen superior se muestra la parte de administración de la web, en la cual se pueden realizar las tareas de administración, donde el usuario administrador podrá eliminar los usuarios que sean necesarios.

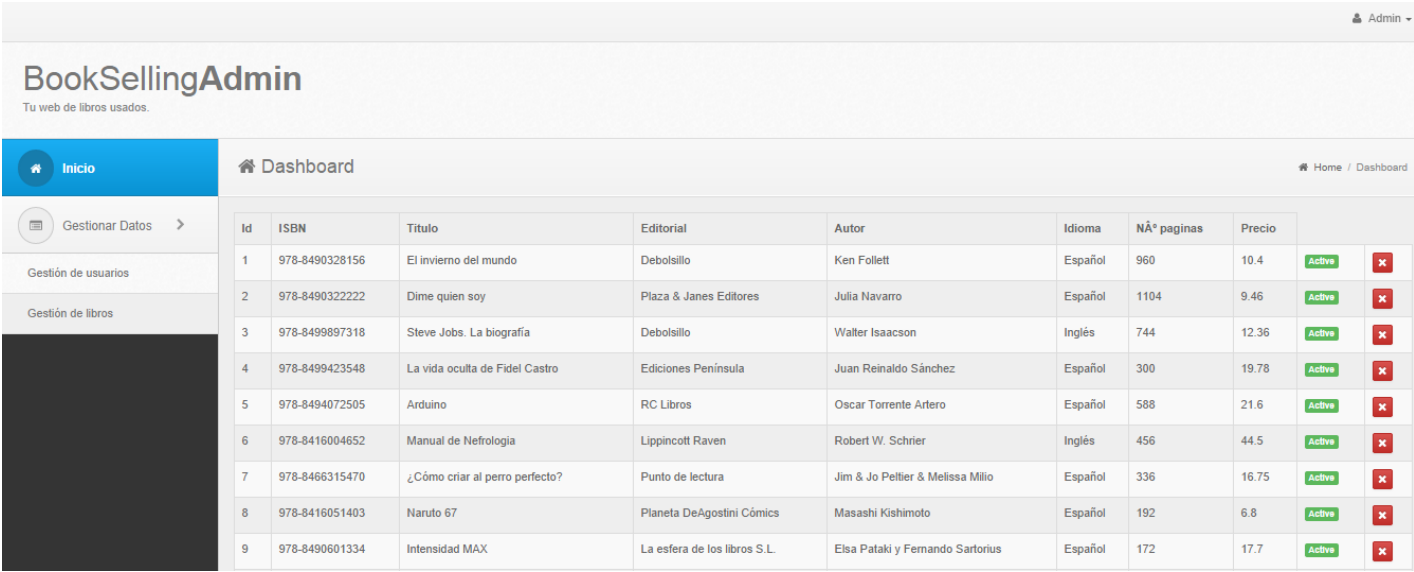


Ilustración 35. Administración. Gestión de los libros.

En esta otra instantánea se puede ver también la parte de gestión de los libros, con la única funcionalidad de eliminar los libros de los usuarios eliminados.

4.3.3. Interfaz: Multi-dispositivo

Como se mencionó en el apartado “2.6.1.Herramientas y tecnologías del cliente”, Bootstrap permite adaptar el contenido de la web a cualquier dispositivo, ajustando el contenido a la resolución de la pantalla del dispositivo desde el que se accede. A continuación se mostrará cómo se visualiza tanto en un Smartphone como en una Tablet:

4.3.3.1. Interfaz Smartphone



Ilustración 36. Interfaz ejemplo en Smartphone

4.3.3.2. Interfaz Tablet

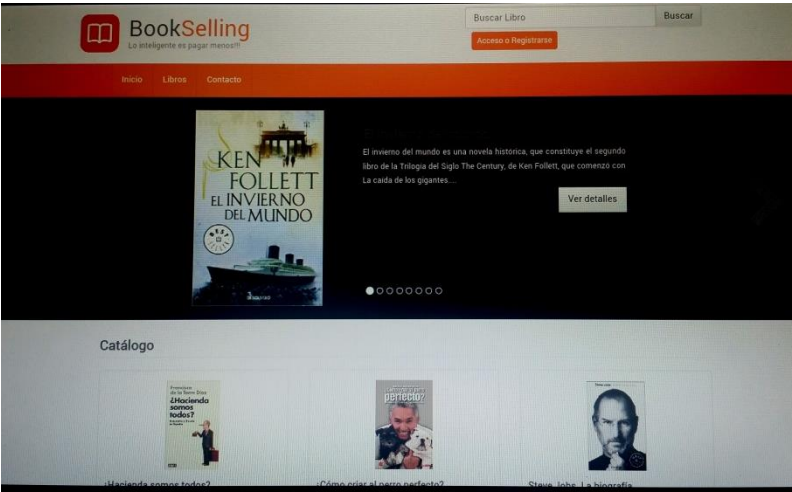


Ilustración 37. Interfaz ejemplo en una tablet.

5. Capítulo 5. Plan de pruebas

Las Pruebas de Software, son una serie de procesos de evaluación del software cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto al cliente. Las pruebas de software son una actividad más en el proceso de calidad.

A continuación se va a especificar un plan de pruebas de aceptación válidas para verificar todos los requisitos establecidos. Todas ellas deben ser superadas para considerar que el proyecto ha finalizado con éxito cumpliendo todos los objetivos establecidos. Finalmente se mostrará una matriz que representará la relación entre los requisitos y las pruebas que verifican su funcionalidad.

5.1 Pruebas de aceptación

El patrón de especificación de las pruebas es el siguiente:

Identificador	
Nombre	
Entrada esperada	Dato o acción de entrada que debe proporcionarse para poder llevar a cabo la prueba
Salida esperada	Resultado de salida que debe obtenerse una vez realizada la prueba
Dependencia	Dependencia existente entre la prueba y los casos de uso.

Tabla 66. Plantilla prueba aceptación.

El identificador será univoco y su abreviatura será PA-XX, siendo XX un número correlativo que se autoincrementa.

PA-01	
Nombre	Registro de usuario
Entrada esperada	El usuario accede al formulario de registro.
Salida esperada	Se muestra un formulario de registro, que el usuario debe rellenar en función de unos valores predeterminados y con el formato exigido. Si el usuario introduce todo correctamente, tras pulsar el botón “Registrarse”, el usuario se habrá registrado
Dependencia	CUU-01

Tabla 67. PA-01. Registro de usuario.

PA-02	
Nombre	Acceso a la web
Entrada esperada	El usuario introduce sus credenciales correctamente, pulsa Entrar”.
Salida esperada	Si todo ha ido bien, en la barra de navegación, aparecerá el nombre usuario que ha accedido a la web, el menú personal de cada usuario y un botón “Salir”.
Dependencia	CUU-02

Tabla 68. PA-02. Acceso a la web.

PA-03	
Nombre	Acceso a los datos personales
Entrada esperada	El usuario accede a la web y en la barra de navegación, accede a su menú personal. Selecciona el apartado "Mis datos personales"
Salida esperada	Deberán aparecer todos los datos personales relacionados con el usuario que se ha conectado a la web.
Dependencia	CUU-09

Tabla 69. PA-03. Acceso a los datos personales.

PA-04	
Nombre	Modificar datos personales
Entrada esperada	El usuario accede a la web y en la barra de navegación, accede a su menú personal. Selecciona el apartado "Modificar datos personales"
Salida esperada	Deberá aparecer un formulario de actualización de los datos personales del usuario conectado en la web.
Dependencia	CUU-10

Tabla 70. PA-04. Modificar datos personales.

PA-05	
Nombre	Subir libros
Entrada esperada	El usuario accede a la web y en la barra de navegación, accede a su menú personal. Selecciona el apartado "Subir libro"
Salida esperada	Deberá aparecer un formulario de subida de libro, el usuario rellenara los campos correctamente y pulsará el botón "Subir".
Dependencia	CUU-13

Tabla 71. PA-05. Subir libros.

PA-06	
Nombre	Historial de libros subidos
Entrada esperada	El usuario accede a la web y en la barra de navegación, accede a su menú personal. Selecciona el apartado "Mis libros"
Salida esperada	Aparecerá una tabla con todos los libros que ha subido el usuario conectado a la web.
Dependencia	CUU-14

Tabla 72. PA-06. Historial de libros subidos.

PA-07	
Nombre	Historial de pedidos
Entrada esperada	El usuario accede a la web y en la barra de navegación, accede a su menú personal. Selecciona el apartado "Mis pedidos"
Salida esperada	Aparecerá una tabla con todos los pedidos que ha hecho el usuario conectado a la web.
Dependencia	CUU-11

Tabla 73. PA-07. Historial de pedidos.

PA-08	
Nombre	Carrito de pedido actual.
Entrada esperada	El usuario accede a la web y en la barra de navegación, accede a su menú personal. Selecciona el apartado "Mi carrito"
Salida esperada	Aparecerá una tabla con todos los libros añadidos al carro por el usuario conectado a la web.
Dependencia	CUU-12

Tabla 74. PA-08. Carrito de pedido actual.

PA-09	
Nombre	Buscar libro
Entrada esperada	El usuario accede a la web y le aparecerá en la cabecera de la web un buscador.
Salida esperada	Un catálogo donde aparecerá el libro buscado si existiese.
Dependencia	CUU-04

Tabla 75. PA-09. Buscar libro.

PA-10	
Nombre	Libros por categorías
Entrada esperada	El usuario accede a la web y navegará hasta la sección "Libros" que aparece en la barra de navegación.
Salida esperada	Una vez en la sección libros, en la barra de navegación vertical de la izquierda, aparecerán una serie de opciones, seleccionando "Categorías", el usuario conectado a la web podrá elegir la categoría de libros a mostrar. Mostrándose un catálogo de libros por la categoría elegida.
Dependencia	CUU-05

Tabla 76. PA-10. Libros por categorías.

PA-11	
Nombre	Top ventas
Entrada esperada	El usuario accede a la web y navegará hasta la sección "Libros" que aparece en la barra de navegación.
Salida esperada	Una vez en la sección libros, en la barra de navegación vertical de la izquierda, aparecerán una serie de opciones, seleccionando "Top ventas". Mostrándose un catálogo de libros vendidos.
Dependencia	CUU-05

Tabla 77. PA-11. Top ventas.

PA-12	
Nombre	Novedades
Entrada esperada	El usuario accede a la web y navegará hasta la sección "Libros" que aparece en la barra de navegación.
Salida esperada	Una vez en la sección libros, en la barra de navegación vertical de la izquierda, aparecerán una serie de opciones, seleccionando "Novedades". Mostrándose un catálogo de libros recientemente añadidos al catálogo.
Dependencia	CUU-05

Tabla 78. PA-12. Novedades.

PA-13	
Nombre	Ver detalles del libro
Entrada esperada	El usuario accede a la web y una vez allí podrá acceder a los detalles de cualquier libro que aparezca en cualquiera de las secciones de la página.
Salida esperada	Aparecerá una imagen de la portada del libro, el precio del libro, los usuarios que han subido el libro para venderlo, así como toda una serie de características del libro en cuestión.
Dependencia	CUU-06

Tabla 79. PA-13. Ver detalles del libro.

PA-14	
Nombre	Añadir libro al carro del pedido
Entrada esperada	El usuario accede a la web y se conecta a la web utilizando sus credenciales. Le aparecerán los libros en cualquiera de los catálogos con un botón "Añadir al carro".
Salida esperada	El libro se añadirá al carro del pedido del usuario conectado a la web.
Dependencia	CUU-15

Tabla 80. PA-14. Añadir libro al carro del pedido.

PA-15	
Nombre	Ordenar libros alfabéticamente
Entrada esperada	El usuario accede a la web y en los catálogos de la misma le aparecerá un botón en la parte superior derecha del catálogo "Ordenar por" con cuatro opciones. Si selecciona "Alfabéticamente ascendente".
Salida esperada	Se mostrará el catalogo ordenado alfabéticamente de forma ascendente (a-z).
Dependencia	CUU-07

Tabla 81. PA-15. Ordenar libros alfabéticamente (a-z).

PA-16	
Nombre	Ordenar libros alfabéticamente
Entrada esperada	El usuario accede a la web y en los catálogos de la misma le aparecerá un botón en la parte superior derecha del catálogo "Ordenar por" con cuatro opciones. Si selecciona "Alfabéticamente descendente".
Salida esperada	Se mostrará el catalogo ordenado alfabéticamente de forma descendente (z-a).
Dependencia	CUU-07

Tabla 82. PA-16. Ordenar alfabéticamente (z-a).

PA-17	
Nombre	Ordenar libros por precio ascendente.
Entrada esperada	El usuario accede a la web y en los catálogos de la misma le aparecerá un botón en la parte superior derecha del catálogo "Ordenar por" con cuatro opciones. Si selecciona "Precio ascendente".
Salida esperada	Se mostrará el catalogo ordenado por precio de menor a mayor.
Dependencia	CUU-07

Tabla 83. PA-17. Ordenar libros por precio ascendente.

PA-18	
Nombre	Ordenar libros por precio descendente.
Entrada esperada	El usuario accede a la web y en los catálogos de la misma le aparecerá un botón en la parte superior derecha del catálogo "Ordenar por" con cuatro opciones. Si selecciona "Precio descendente".
Salida esperada	Se mostrará el catalogo ordenado por precio de mayor a menor.
Dependencia	CUU-07

Tabla 84. PA-18. Ordenar libros por precio descendente.

PA-19	
Nombre	Eliminar libro del carrito.
Entrada esperada	El usuario accede a la web y en la barra de navegación, accede a su menú personal. Selecciona el apartado "Mi carrito"
Salida esperada	Aparecerá una tabla con todos los libros añadidos al carro por el usuario conectado a la web. Podrá eliminar del carrito el libro o los libros que desee.
Dependencia	CUU-16

Tabla 85. PA-19. Eliminar libro del carrito.

PA-20	
Nombre	Envío de mensaje de contacto
Entrada esperada	El usuario entra en la web y accede a la sección “Contacto” de la barra de navegación superior.
Salida esperada	Aparece un formulario donde el usuario puede rellenar para enviar a la web preguntando cualquier duda o consulta.
Dependencia	CUU-17

Tabla 86. PA-20. Envío mensaje de contacto.

PA-21	
Nombre	Salir de la web
Entrada esperada	El usuario podrá cerrar su sesión y salir de la web. Pulsando el botón "Salir".
Salida esperada	Aparecerá la visión de la página de cualquier usuario no registrado en la web o no conectado.
Dependencia	CUU-18

Tabla 87. PA-21. Salir de la web.

A continuación se mostrará una matriz que representará la relación entre los requisitos de software funcionales y las pruebas que verifican su funcionalidad, para que se asegure que todos los requisitos están recogidos en las pruebas realizadas.

	PA-01	PA-02	PA-03	PA-04	PA-05	PA-06	PA-07	PA-08	PA-09	PA-10	PA-11	PA-12	PA-13	PA-14	PA-15	PA-16	PA-17	PA-18	PA-19	PA-20	PA-21
RSF-1	X																				
RSF-2		X																			
RSF-3		X	X																		
RSF-4		X		X																	
RSF-5		X			X																
RSF-6		X				X															
RSF-7		X					X														
RSF-8		X						X													
RSF-9									X												
RSF-10										X											
RSF-11											X										
RSF-12												X									
RSF-13													X								
RSF-14	X													X							
RSF-15															X						
RSF-16																X					
RSF-17																	X				
RSF-18																		X			
RSF-19	X																		X		
RSF-20																				X	
RSF-21																					X

Tabla 88. Matriz de trazabilidad. Prueba aceptación – Requisito funcional

5.1.1 Resultado de la pruebas

Una vez finalizado el desarrollo de la aplicación y pasado la batería de pruebas se observa que los resultados han sido satisfactorios, la ejecución de todas las pruebas se ha llevado a cabo de forma correcta. La tabla que representa los resultados obtenidos en dichas pruebas es la siguiente:

Prueba	Resultado
PA-01	Satisfactorio
PA-02	Satisfactorio
PA-03	Satisfactorio
PA-04	Satisfactorio
PA-05	Satisfactorio
PA-06	Satisfactorio
PA-07	Satisfactorio
PA-08	Satisfactorio
PA-09	Satisfactorio
PA-10	Satisfactorio
PA-11	Satisfactorio
PA-12	Satisfactorio
PA-13	Satisfactorio
PA-14	Satisfactorio
PA-15	Satisfactorio
PA-16	Satisfactorio
PA-17	Satisfactorio
PA-18	Satisfactorio
PA-19	Satisfactorio
PA-20	Satisfactorio
PA-21	Satisfactorio

Tabla 89. Resultado de las pruebas.

6. CAPÍTULO 6 GESTIÓN DEL PROYECTO

En este capítulo se desarrollara todo lo relativo a la gestión del proyecto. De este modo, se podrá ver la planificación inicial del proyecto y los diferentes cambios que se han producido a los largo de su ciclo de vida, mediante una comparativa de una planificación final. Además se definirán los recursos que han sido necesarios para la realización del proyecto.

6.1 Planificación del proyecto

En este primer apartado se detallarán todas las tareas realizadas a lo largo del proyecto y se realizará una comparación entre la planificación inicial y final del mismo tomando como referencia el diagrama de Gantt elaborado en la primera fase.

6.1.1 Planificación inicial

En primer lugar se encuentra la tarea de planificación. En ella se realizó la estimación de los días de trabajo que iban a transcurrir con la realización de cada tarea hasta la fecha límite (Dicha fecha se ha considerado el 6 de Agosto). Cabe destacar que es en este primer bloque de tareas cuando se realizó el diagrama de Gantt mostrado posteriormente.

Más tarde daría comienzo el conjunto de tareas relacionadas con el análisis. En esta parte se planificaron las tareas de análisis de aplicaciones similares a la desarrollada (esto del arte), que estaban encaminadas a proporcionar la información necesaria para conocer si el desarrollo de la aplicación supondría algo innovador y útil en cuanto a las aplicaciones ya existentes se refiere. De la misma forma, se planificaron las tareas relacionadas con el análisis de alternativas y, por último, la obtención de requisitos.

Posteriormente, da comienzo la fase de diseño de la aplicación. En dicha fase se planifica analizar la arquitectura del sistema, identificando el tipo de aplicación a desarrollar y los componentes que la forman.

Mientras el proceso de diseño es llevado a cabo, se planifica implementar y codificar la aplicación. Se planificó en primer lugar la instalación del entorno de programación y el respectivo aprendizaje del lenguaje a utilizar.

Finalmente, se realizó la estimación de la fase de pruebas y la comprobación de los últimos errores y retoques. Todos los conjuntos de tareas descritos anteriormente serán realizados de forma paralela a la presente documentación, completando los diferentes apartados del documento desde el primer día de trabajo.

El diagrama de Gantt elaborado que refleja la planificación de las tareas mencionadas es el siguiente:

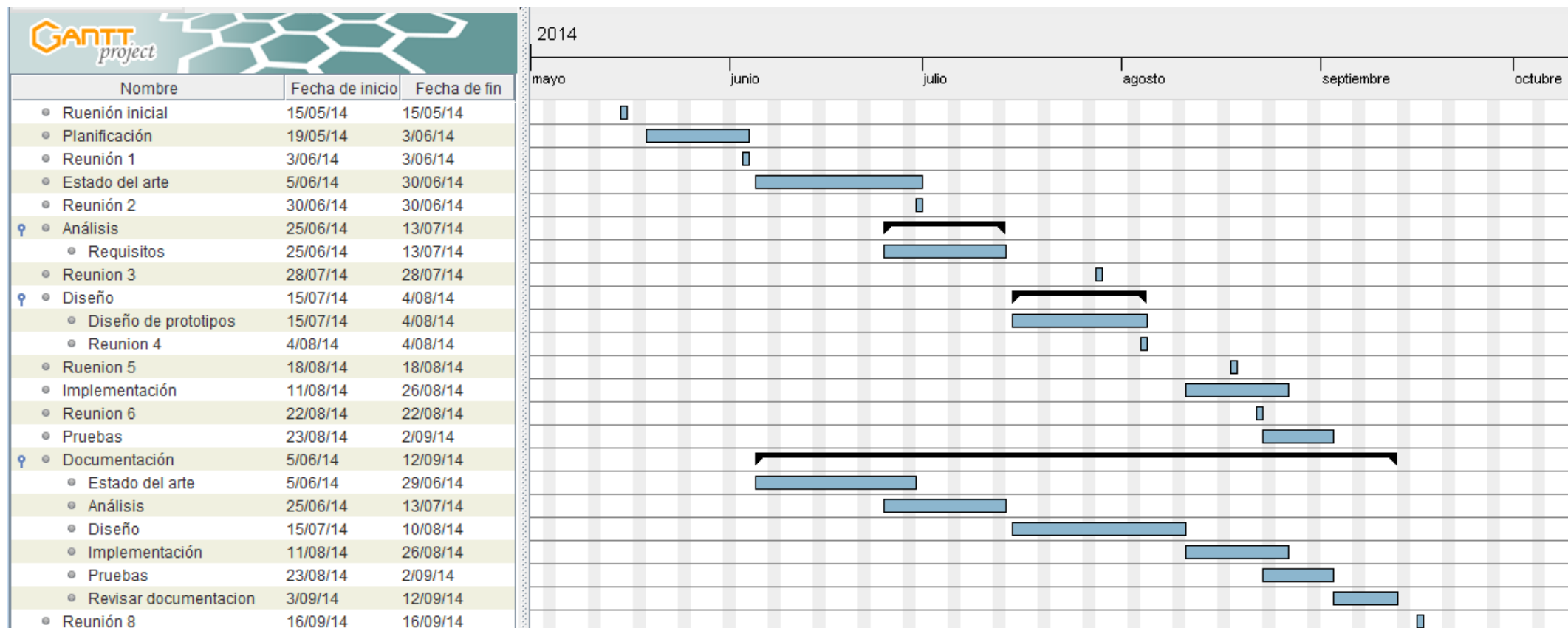


Ilustración 38. Planificación inicial del proyecto.

Cabe destacar que se han realizado más reuniones de seguimiento aparte de las mencionadas en el diagrama anterior. No se han constatado en el diagrama puesto que fueron reuniones informales, de consulta a través de videoconferencia.

6.1.2 Planificación final

La planificación inicial ha sufrido varios cambios a lo largo del desarrollo del proyecto. Esto ha supuesto un cambio significativo, re planificando el tiempo de trabajo dedicado a varias de las tareas planificadas. Dichos cambios y los factores que han intervenido para variar la planificación inicial han sido los siguientes:

La fase de planificación se ha completado según previsto, realizando las tareas en el tiempo de trabajo especificado. Por otro lado, la fase de análisis ha sufrido algún que otro cambio, pero cumpliendo con los plazos establecidos.

El cambio significativo que ha sufrido esta planificación ha empezado en Abril, ya que en un principio se pensó que se seguiría trabajando con el mismo ritmo, pero este fue interrumpido por la carga de trabajo de las asignaturas del último cuatrimestre de la carrera.

Todo esto ha supuesto un retraso en todas las tareas a partir del análisis. La pérdida de este mes en mitad del desarrollo del proyecto ha supuesto a parte de un retraso lógico de un mes de finalización del mismo ciertos problemas con la compatibilidad de dicho proyecto con el estudio de exámenes finales del cuatrimestre.

Aun con todos estos inconvenientes y el retraso se ha cumplido con los plazos esperados tanto en diseño como en implementación, únicamente que se ha desplazado un mes. En cuanto al plan de pruebas se ha realizado en el menor tiempo posible para poder organizar todos antes de la entrega del día 5 de septiembre.

Finalmente, la fase de documentación ha sido la que menos cambios en cuanto a la planificación inicial ha sufrido. Esto es debido a que la redacción del presente documento diera comienzo el primer día de trabajo, tal y cómo se refleja en la planificación inicial. Y destacar que las reuniones han sido muy parecidas a las previstas.

A continuación se muestra las tareas y el diagrama de Gantt que refleja la planificación final del proyecto:

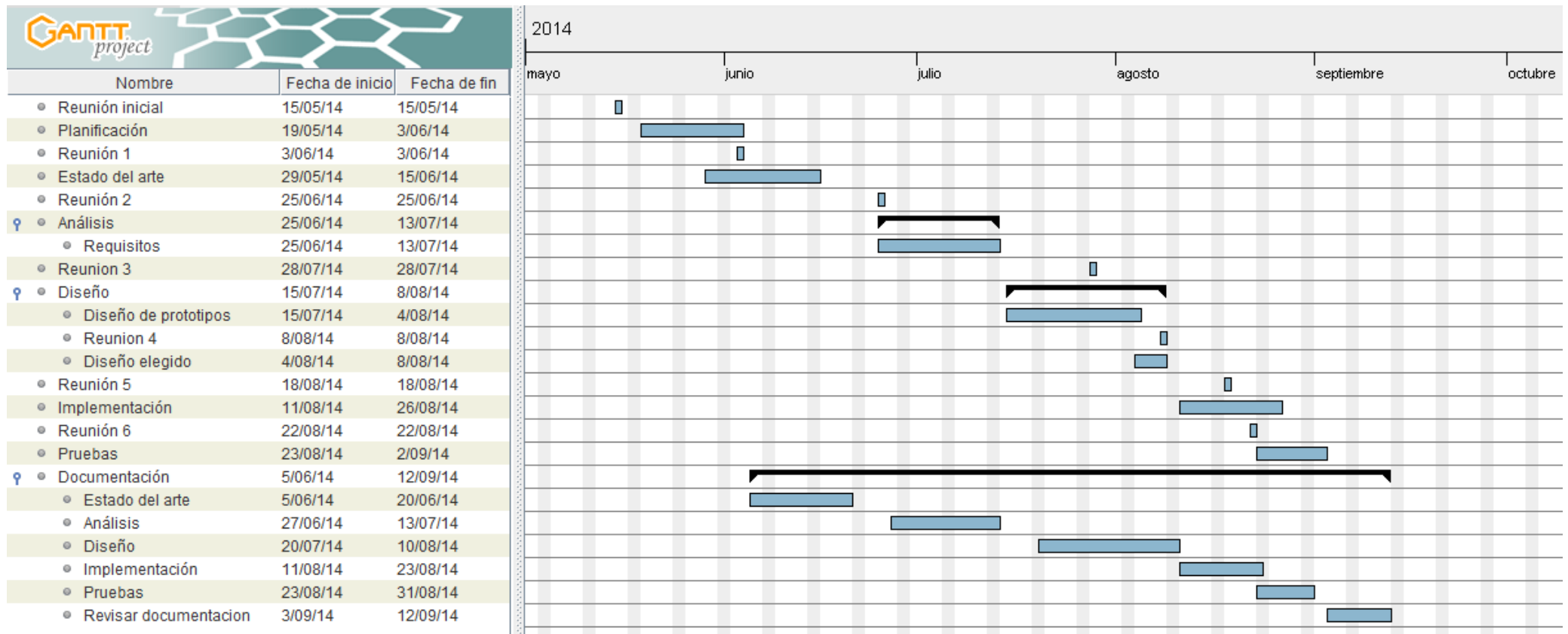


Ilustración 39. Planificación final del proyecto.

6.2 Análisis económico

Una vez conocido el esfuerzo necesario para la realización del proyecto, se pasa a detallar el coste del mismo. En primer lugar se computan los gastos asociados al personal que se encarga de la realización de la aplicación. Según el cargo que ocupe su salario varía. Tras esto se explica detalladamente los gastos de hardware y de software necesarios para la realización del proyecto. Además se contabilizan también otros tipos de gastos que conlleva realizar todas las acciones correspondientes al proyecto, como por ejemplo la luz empleada en los sistemas para el funcionamiento. Para finalizar, se contabilizará todos los gastos y se dará una cifra exacta del precio de la aplicación.

6.2.1 Coste personal

Como personal se ha incluido a Jorge Alonso López como principal desarrollador de la aplicación. También a Jesús Hernando Corrochano, ya que es el tutor del proyecto y se deben computar sus horas de reuniones y ayuda en el desarrollo de la aplicación.

A continuación se mostrará una tabla con todos los detalles:

Personal	Categoría	Coste hombre/hora	Dedicación (horas)	Coste (€)
Alonso López, Jorge	Ingeniero Junior	30	267	8.010
Hernando Corrochano, Jesús	Ingeniero Senior	60	25	1.500
Hombre/hora		90	Total	9.510

Tabla 90. Detalles coste personal.

Para calcular las horas del Ingeniero Junior se ha establecido que a lo largo de la realización del proyecto se han llevado a cabo tres horas diarias. El coste mencionado ya incluye los valores del IRPF y Seguridad Social.

El coste asociado al personal es de **9.510 euros**.

6.2.2 Coste del Hardware

Concepto	Unidades	Precio Unitario(€)	Vida útil estimada	Tiempo de uso (meses)	Coste para el proyecto
Ordenador MacBookAir 13"	1	928,35	48	2	38,68
Toshiba Satellite Z30	1	875,65	48	3	54,73
Total					93.41

Tabla 91. Coste de hardware.

6.2.3 Coste del software

Concepto	Unidades	Precio Unitario (euros)	Coste para el proyecto (€)
MAC OS X Mountain Lion 10.6.8	1	0	0
Windows 8	1	0	0
Microsoft Office hogar y estudiantes 2010	1	99	99
GitHub	1	0	0
GanttProject2.6.6	1	0	0
MySQL Workbench Version Open Source	1	0	0
Sublime Text	1	0	0
Node	1	0	0
MongoDb	1	0	0
Total			99

Tabla 92. Coste del software.

6.2.4 Costes totales

Concepto	Coste total(€)
Personal	9.510
Hardware	93,41
Software	99
Total	9.702,41

Tabla 93. Costes totales.

Finalmente puede concluirse que el coste total del proyecto asciende a NUEVE MIL SETECIENTOS DOS CON CUARENTA Y UN EUROS (9.702,41 €) IVA no incluido.

7. CAPÍTULO 7: CONCLUSIONES

A continuación se disponen las conclusiones obtenidas una vez finalizado el proyecto. En primer lugar se establecerán las conclusiones relacionadas con el proyecto en sí, posteriormente las conclusiones a nivel personal y para finalizar las líneas futuras para la ampliación y mejora de la aplicación.

7.1. Conclusiones del proyecto

Se ha desarrollado como trabajo fin de grado una aplicación web en la cual los usuarios que se registraran en ella podían vender y comprar libros usados.

El desarrollo de este trabajo fin de grado ha supuesto dos problemas destacados. El primero de ellos es el diseño de interfaces sencillas y atractivas. El segundo de ellos ha sido la creación del servidor donde ejecutar la aplicación web.

El primer problema ha supuesto marcar nuevos objetivos como informático. A lo largo de la carrera en escasas ocasiones se ve cómo se diseña una interfaz. En ningún caso nos enseñan cómo crear una interfaz atractiva para el usuario. Este hecho supuso realizar una investigación acerca de cómo solucionar este problema. El diseño de las interfaces ha requerido la comprensión y el manejo de las herramientas y características de Bootstrap.

El segundo problema implica la preparación de un servidor donde ejecutar la aplicación web, esto trajo consigo muchos problemas al principio dado que nunca jamás a lo largo de la carrera había realizado una tarea así, por lo que tuve que leer documentación y realizar varios tutoriales, así como videos explicativos, para poder llevar a cabo dicha tarea. A pesar de que al principio costó un poco, logre llevar a cabo mi propósito y pude crear mi servidor donde ejecutar mi web. Para controlar todos los posibles casos se ha hecho hincapié en pruebas de aceptación. Todo ello para intentar quebrar el correcto funcionamiento de la aplicación.

Otro problema que ha ido sucediendo a la hora de todo el proceso, ha sido la constante aparición de requisitos nuevos, y al trabajar con desarrollo en cascada con requisitos no cerrados, un avance lento, ya que se debía volver al principio, pasando por todas las fases. Esto me hizo darme cuenta de lo importante que es elegir una correcta metodología de desarrollo de software, ya que facilita mucho el trabajo a lo largo de todo el proyecto. Si tuviera que escoger de nuevo la metodología, elegiría una metodología ágil, más adaptada a los cambios durante el proceso de desarrollo del software.

La planificación inicial se ha visto postergada debido a factores que han aparecido a lo largo del desarrollo. A medida que se iba aprendiendo el lenguaje de programación⁹ se iba cogiendo confianza y descubriendo nuevas funcionalidades y métodos de hacer las cosas de una manera mejor¹¹. Se ha aprendido a desarrollar en un entorno nuevo y desconocido, a investigar mejores maneras de programar... en definitiva, se han adquirido las habilidades necesarias para ser capaz de desarrollar un proyecto de manera eficaz.

En este proceso no todo han sido éxitos. Ha habido momentos en los que han surgido problemas. La mayor parte de ellos se han solucionado de una forma u otra. Pero sobre todo

mencionar los problemas surgidos al utilizar un desarrollo en cascada, es que al aparecer requisitos nuevos, todo debía ser rediseñado desde un principio, afectando al avance general del proyecto. He podido comprobar en primera persona los problemas que surgen utilizando este tipo de metodologías y valorar lo ofrecido por otras que se eligieron.

En lo que respecta a los objetivos que se habían establecido para este TFG, se evaluará hasta qué punto se ha llegado a la consecución de los mismos. Para ello, se volverán a enumerar cada uno de ellos, adjuntando las conclusiones que se pueden obtener tras el desarrollo:

- Debe ser una tienda online. Este objetivo es la esencia de la aplicación web desarrollada. Se basa en crear una tienda de compra-venta de libros usados por parte usuarios registrados en la web. Realizando el pedido entre ellos, siendo la web la intermediaria entre ellos.
- A ser posible que sea original. Este objetivo se cree que se ha cumplido gracias al empeño y dedicación puestos en el diseño. Se cree que es de lo más original y más orientado al público que va destinado esta aplicación.

Para finalizar comentar que se ha iniciado todo desde cero y se ha pasado por todas las fases de desarrollo para concluir la creación del proyecto. Muchos de los conocimientos adquiridos a lo largo de la carrera se han puesto de manifiesto al afrontar su desarrollo. Por tanto cabe decir que es una gran satisfacción ver cómo una idea puede llevarse a cabo si se trabaja con dedicación y con empeño.

7.2. Líneas futuras

Una vez expuestas las conclusiones, se pueden obtener una serie de líneas futuras que no se extraen de los objetivos pero que sería interesante desarrollar. Estas pueden mejorar las prestaciones de la aplicación para una mejor aceptación en el mercado.

Al tratarse de una tienda online el objetivo es la compra del producto cuando accedes a ella, pero realmente lo que se ha implementado ha sido el pedido entre dos personas físicas que utilizan la aplicación web como intermediaria. Como propósito futuro sería implementar una pasarela de pago PayPal que hiciese que cada usuario pudiera formalizar el pago.

Otra idea también sería gestionar y planificar el tema de los envíos de los artículos a los usuarios, esta idea conlleva en diseñar e implementar una aplicación de seguimiento de los paquetes, que son enviados a través de una empresa de logística.

Otra posible mejora sería la creación de bonos para comprar en la web, en los cuales los usuarios registrados pudieran realizar la compra de bonos para gastar en libros, obteniendo algún tipo de descuento por utilizar dichos bonos.

Para adquirir un gran catálogo de libros online, se tiene en mente asociarse con librerías que tienen exceso de libros usados acumulados o incluso nuevos descatalogados y que quieren deshacerse de ellos, podrían añadirlos a la web y así crear una gran tienda de libros online con muchos títulos disponibles.

Todas estas mejoras y todo lo que está implementado me gustaría que tras el uso e investigación de los usuarios accediendo en ella, recibiera la opinión de lo mejor y lo peor de la aplicación, para realizar mejoras que satisfagan las necesidades de los usuarios.

Por ultimo otra idea futura sería abrir el campo más allá de las fronteras de España y añadir otro idioma para la interfaz de la web, posibilitando a que todo el mundo la pudiera utilizar sin tener problemas con el idioma, ya que el inglés es el idioma internacional con el que todos nos comunicamos mundialmente.

8. BIBLIOGRAFÍA

- [1] Abebooks: http://www.iberlibro.com/vender-libros/?cm_sp=TopNav-_-Home-_-Sell
- [2] Agile Alliance: <http://www.agilealliance.org/>
- [3] Amazon: http://www.amazon.es/b/ref=nav_topnav_sell?ie=UTF8&node=2383605031
- [4] Apache Tomcat: <http://www.mysql.com/>
- [5] Bootstrap: <http://getbootstrap.com/>
- [6] Casa del libro: <http://www.casadellibro.com/ayuda/segundaMano>
- [7] CSS3: http://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada
- [8] Comercio electrónico:

Drucker, P. F. (2001). Detrás de la revolución de la información. *Revista La Factoría*, octubre-enero, (13).
- [9] Desarrollo en cascada: http://es.wikipedia.org/wiki/Desarrollo_en_cascada
- [10] Eclipse: <https://www.eclipse.org/>
- [11] GantProject: <http://www.ganttproject.biz/>
- [12] GitHub: <https://netbeans.org/>
- [13] Glassfish: <https://glassfish.java.net/es/>
- [14] HTML5: <http://www.blogeninternet.com/2013/04/que-es-html5-y-para-que-sirve.html>
- [15] JavaScript: <http://es.wikipedia.org/wiki/JavaScript>
- [16] JavaScript II:

Flanagan, D. (2007). *JavaScript. La Guía Definitiva*.

[17] JEE: http://es.wikipedia.org/wiki/Java_EE

[18] Java Enterprise Edition:

Groussard, T. (2010). *Java Enterprise Edition: Desarrollo de aplicaciones web con JEE 6*.

Ediciones ENI.

[19] JQuery: <http://jquery.com/>

[20] Manual de JQuery:

Alvarez, M. A. (2010). Manual de jQuery. *Recuperado el, 17*.

[21] Metodologías ágiles: http://es.wikipedia.org/wiki/Manifiesto_%C3%A1gil

[22] MongoDB: <http://www.mongodb.org/>

[23] MySQL: <http://www.mysql.com/>

[24] NetBeans: <https://netbeans.org/>

[25] Node características: <http://www.desarrolloweb.com/articulos/caracteristicas-nodejs.html>

[26] Node.js: <http://nodejs.org/api/>

[27] Rational Software Architect: <http://www-03.ibm.com/software/products/es/ratisoftarch>

[28] RUP:

Martínez, A., & Martínez, R. (2002). *Guía a Rational Unified Process*.

[29] Socket IO: <http://socket.io/>

[30] Sublime Text 3: http://es.wikipedia.org/wiki/Sublime_Text

[31] Twig: <http://twig.sensiolabs.org/>

[32] Tutorial de Node.js: <http://www.nodebeginner.org/index-es.html>

[33] Vender por internet artículos de segunda mano:

<http://www.lynkoo.com/legal/vender-por-internet-productos-de-segunda-mano/>

[34] WebSphere Application Server: <http://www-03.ibm.com/software/products/es/appserv-was>

[35] Why buy? Historia del E-commerce: <http://culturacolectiva.com/la-historia-del-e-commerce/>